DEVELOPING A TRAINING PROGRAM FOR THE TRAFFIC ALERT AND COLLISION AVOIDANCE SYSTEM IN CONTEXT

A Thesis
Presented to
The Academic Faculty

by

Elizabeth S. Fleming

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Aerospace Engineering

Georgia Institute of Technology
May 2013

DEVELOPING A TRAINING PROGRAM FOR THE TRAFFIC ALERT AND COLLISION AVOIDANCE SYSTEM IN CONTEXT

Approved by:

Dr. Amy Pritchett, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Karen Feigh
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Ute Fischer
School of Literature, Communication, & Culture
Georgia Institute of Technology

Date Approved: February 8th, 2013

ACKNOWLEDGEMENTS

Dr. Amy Pritchett, thank you for introducing me to cognitive engineering and giving me the opportunity to design this research project. I would also like to thank my committee for their support and guidance. Dr. Ute Fischer, thank you for helping me with the initial development of this study. And, Dr. Karen Feigh, thank you for teaching me about statistical methods.

I would also like to thank Dr. Wesley Olson with MIT Lincoln labs for providing wonderful insight about TCAS. Also, thank you to Wayne Gallo and Roger Sultan for reviewing the training program and giving feedback about the training program design.

Thank you to the many members of the TCAS team, specifically William Cleveland, Vlad Popescu, Justin Mullins, Henry Tran, Jack Ridderhof, Alyssa Whitlock, and Colin Ludwig for helping to staff my experiment. Thank you to my undergraduate research assistant, Kylie Garey, who developed the initial Matlab code for analyzing the data. Also, thank you to Anil Bozan who read through my thesis and provided wonderful feedback.

In addition, I would like to thank Alexandra Coso and Zarrin Chua for their continued support, encouragement, and caffeine. Thank you to my parents, who acted as personal cheerleaders throughout my undergraduate career and have continued to provide support as I progress through graduate school.

Finally, a huge thank you to TJ Lindsley, who has endured many evenings listening to me rattle on about my research. TJ, thank you for always supporting my endeavors and know that I will eventually graduate from college.

I would like to acknowledge that this material is based upon work supported by a Cooperative Agreement (DTFAWA-10-C-00084) with the Federal Aviation Administration (FAA) Human Factors Research and Engineering Group, with Tom McCloy acting as technical manager. Additionally, I am supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0644493.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ASRS	Aviation Safety Reporting System
ATC	Air Traffic Control
CEC	Cognitive Engineering Center
DBT	Demonstration Based Training
EBT	Event Based Training
FAA	Federal Aviation Administration
GA	General Aviation
IFR	Instrument Flight Rules
KBB	Knowledge-Based Behavior
LOFT	Line Oriented Flight Training
NMAC	Near Mid-Air Collision
PFD	Primary Flight Display
RA	Resolution Advisory
RBB	Rule-Based Behavior
RFS	Reconfigurable Flight Simulator
SBB	Skill-Based Behavior
TA	Traffic Advisory
TCAS	Traffic Advisory and Collision Avoidance System
TOPA	TCAS Operational Performance Assessment
TSD	Traffic Situation Display
VFR	Visual Flight Rules
VSI	Vertical Speed Indicator

SUMMARY

The Traffic alert and Collision Avoidance System (TCAS) is an aircraft collision avoidance system designed to prevent mid-air collisions. During an advisory, danger is imminent, and TCAS is assumed to have better, more up-to-date information than the ground operated air traffic control (ATC) facility. Following a TCAS RA is generally the safe course of action during an advisory. However, pilot compliance with RAs is surprisingly low. Results from a TCAS monitoring study show pilots are not complying with many TCAS advisories. As revealed by pilot-submitted Aviation Safety Reporting System (ASRS) reports, this noncompliance could be attributed, in part, to pilot confusion to TCAS operation as well as misunderstandings of the appropriate response to a TCAS issued advisory.

This thesis details the development and evaluation of a TCAS training program intended to improve pilots' understanding of TCAS use for collision avoidance in a range of traffic situations. The training program integrated Demonstration Based and Event Based Training techniques. Its efficacy was analyzed in an integrated ATC-cockpit simulator study in which eighteen commercial airline pilots were asked to complete the TCAS training program and afterwards experienced twelve experimental traffic events. The trained pilots' performance was compared to the performance of 16 baseline pilots who did not receive the modified training.

Overall, the training program did have a significant impact on the pilots' behavior and response to TCAS advisories. The measure Time Pilots First Achieved Compliance decreased with the trained pilots, as did the measure Autopilot Disconnect Time After

RA Initiation. Trained pilots exhibited less aggressive performance in response to a TCAS RA (including a decrease in the measures Altitude Deviation Over Duration Of RA, Average Vertical Rate Difference, Maximum Vertical Rate Difference, and Maximum Vertical Rate). The measure Percent Compliance did not significantly vary between trained and baseline pilots, although trained pilots had a more consistent response in the traffic event with conflicting ATC guidance. Finally, on the post-experiment questionnaires, pilots commented on their increase in understanding of TCAS as well as an increase in their trust in the advisory system.

Results of this research inform TCAS training objectives provided by the FAA as well as the design of TCAS training. Additionally, conclusions extend more broadly to improved training techniques for other similar complex, time-critical situations.

CHAPTER 1

INTRODUCTION

The Traffic alert and Collision Avoidance System (TCAS) is an aircraft collision avoidance system designed to prevent mid-air collisions. TCAS monitors the nearby airspace for other aircraft equipped with transponders and displays them on a horizontal, plan-view traffic situation display (TSD). As an additional precaution, TCAS issues advisories that act as a last-resort safety net. Should a near-mid-air collision (NMAC) trajectory arise between two aircraft a “traffic advisory” (TA) is issued in the form of an aural “Traffic Traffic” and is also shown on the TSD. If warranted, a more time-critical “resolution advisory” (RA) is subsequently generated portraying an advised collision avoidance maneuver to the pilot.

During an advisory, danger is imminent, and TCAS is assumed to have better, more up-to-date information than the ground operated air traffic control (ATC) facility. Following a TCAS RA is generally the safe course of action during an advisory. Despite these safety measures and potential loss of life, pilot compliance with RAs is surprisingly low. Results from a TCAS monitoring study show pilots not complying with many TCAS advisories. As revealed by pilot-submitted Aviation Safety Reporting System (ASRS) reports, this noncompliance could be attributed, in part, to pilot confusion to TCAS operation as well as misunderstandings of the appropriate response to a TCAS issued advisory.

Pilot understanding of collision avoidance, specifically using TCAS, is promoted through a variety of training materials and methods. Federal Aviation Administraion's Advisory Circular 120-55C lists several mandated training objectives that should be included in TCAS training programs (Federal Aviation Administration [FAA], 2011a). The document also divides TCAS training into two phases. The first phase is a computer-based training program the pilot completes individually. This initial phase is intended to inform pilots of TCAS equipment, operation, advisories, and limitations. However, preliminary observations from a TCAS experiment recently conducted by the Cognitive Engineering Center (CEC) at the Georgia Institute of Technology have shown instances of pilots not understanding TCAS advisories. The second TCAS training phase occurs in a simulator training flight. However, in informal interviews pilots reported that TCAS training is separated from the realistic complex operational environment: Pilots encountered predictable TCAS advisories with only one target (or perhaps two targets) shown on the TSD. Although approved by the FAA, current training methods used by airlines for TCAS not only lead to pilot misunderstandings about general TCAS operation and advisories, but also do not fully prepare pilots for the context of real advisories.

The first objective of this thesis is to *train pilots to understand TCAS use for collision avoidance in the actual traffic and operational traffic environment*. The training program should support pilots' understanding of TCAS components, operation, and advising logic. Also, TCAS use in the actual environment will be depicted using annotated graphical demonstrations. These demonstrations are intended to give pilots a foundation in their knowledge-based understanding of TCAS and encourage appropriate use of the advisories.

The second objective is to *provide pilots with a well-rounded knowledge of different traffic situations that may result in TCAS advisories*. In addition to the demonstrations, this training also then requires pilots to experience a range of TCAS advisories in an ATC-cockpit integrated simulator. This event based approach is intended to strengthen pilots' understanding of TCAS use in their environmental context while also proving an opportunity to respond to a variety of traffic events in a realistic ecology, including representative target aircraft trajectories and ATC communications.

The organization of this thesis is to first present relevant literature on collision avoidance and training program design. The next chapter discusses the design decisions that were made in the design of an integrated DBT and EBT training program for TCAS. The experimental design and results are outlined using Rassmussen's skills, rules, knowledge as a guiding theme. Finally, the research completed for this thesis is summarized and implications are discussed.

This thesis discusses the development of a TCAS training program intended to improve pilots' understanding of TCAS use for collision avoidance in a range of traffic situations. Results from this thesis may also extend more broadly to improved training techniques for other similar complex, time-critical situations.

CHAPTER 2

LITERATURE REVIEW

The literature review discusses both TCAS development and operation, and research on training program design. Throughout, a range of skills, rules, and knowledge pilots need to effectively interact with TCAS are noted, and then appropriate training mechanisms are identified.

2.1 The Traffic Alert and Collision Avoidance System

The Traffic alert and Collision Avoidance System (TCAS) is an aircraft collision avoidance system designed to reduce the number of mid-air collisions between aircraft. The TCAS unit on an aircraft monitors the nearby airspace for other aircraft equipped with transponders. These other aircraft are shown to the pilot on the horizontal, plan-view traffic situation display (TSD). The TSD, as seen in Figure 1, also shows altitude in text beside the aircraft symbol (e.g. -5 is interpreted as 500 feet below) and an up or down arrow is also shown when the target is climbing or descending. The main purpose of the TSD is to help pilots visually acquire aircraft in the vicinity: the TSD is not considered to have sufficient information to serve as the sole basis for pilot collision avoidance maneuvers (FAA, 2011b).

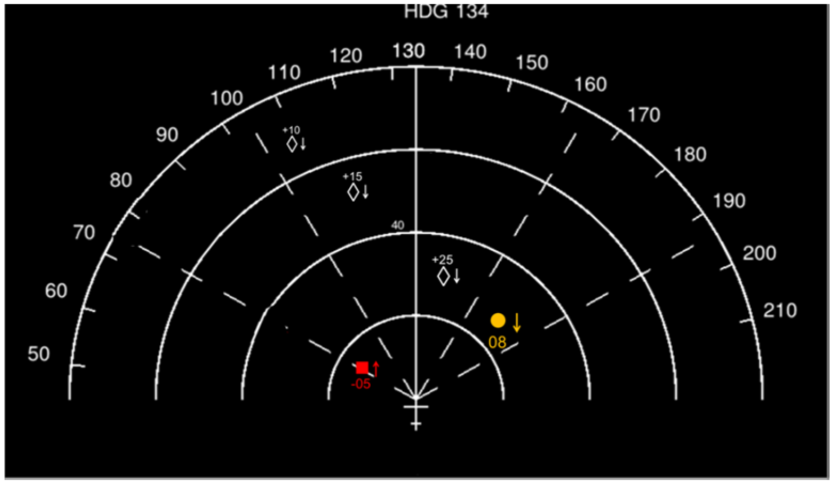


FIGURE 1. TCAS Traffic Situation Display, depicting both a TA and an RA

In addition to the TSD, TCAS issues advisories that act as a last-resort safety net. Should a near-mid-air collision (NMAC) trajectory arise between two aircraft, a “traffic advisory” (TA) is issued in the form of an aural “Traffic Traffic” and on the TSD the target aircraft’s symbol changes to yellow or amber circle. If warranted, a more time-critical “resolution advisory” (RA) is subsequently generated portraying a collision avoidance maneuver to the pilot. RA’s are delivered aurally and visually; for example, if TCAS issues an active RA to climb the pilot hears “Climb, climb” and a positive vertical speed is shown on the primary flight display (PFD) or vertical speed indicator (VSI). Also on the TSD, the target aircraft’s symbol changes to a red square. After the pilot has passed the target aircraft TCAS announces “Clear of conflict” and the pilot is expected to return to the original flight path. If both aircraft in the traffic event are TCAS-equipped, the RA will be coordinated between the two TCAS units; that is, if aircraft A receives a “Climb” RA, aircraft B will received a “Descend” RA (FAA, 2011b).

However, many of the TCAS advisories are triggered by aircraft not equipped with TCAS. For example, the TA or RA may be triggered by traffic flying under “Visual Flight Rules” (VFR). VFR traffic may or may not be communicating with ATC and is permitted to fly 500 feet below other traffic. The VFR pilot may or may not be aware of neighboring aircraft; even when aware, the pilot is following an allowable flight path and, if not equipped with TCAS, will not know that his or her path has caused another aircraft to be given a TCAS advisory. The TCAS Operational Performance Assessment (TOPA) program found that in 91% of RA’s recorded, one of the two aircraft are not equipped with TCAS. (Olszta & Olson, 2011; Olson & Olszta, 2010).

Pilot compliance to an RA is mandated by the Federal Aviation Administration (FAA) Advisory Circular 120-55C unless the pilot believes that the maneuver would endanger safe flight operations (FAA, 2011a; FAA 2003). However, the time period leading up to and spanning a TCAS RA, may also include many other events. For instance, a pilot may receive a traffic call-out from the controller or may visually acquire another aircraft (which may or may not be the advised traffic). In the same instance, the pilot may overhear other communications on the party-line or experience non-collision avoidance related events and alerts. All of these other events may impact the pilot’s belief in the TCAS advisory.

To prepare pilots to respond to TCAS advisories, training standards for TCAS, as outlined in FAA Advisory Circular 120-55C, explicitly separate the training requirements into two segments: ground-based training requirements and flight training requirements. This type of division is common in aviation training and aligns with training for other systems (FAA, 2011a). The ground-training requirements are often met through classroom instruction, but recently the FAA has permitted online and computer-based training to take the place of some classroom training.

The areas that must be covered in ground training are primarily related to general concepts of TCAS and its operation. These standards highlight the need for pilots to understand the types of advisories TCAS delivers as well as how those advisories are generated. Other standards call for training on the limitations of TCAS and ground operation procedures (FAA, 2011a).

TCAS flight training standards require that pilots be provided the opportunity to maneuver in response to a TCAS advisory at some point in their simulator training. Typically, TCAS flight training is integrated with line-oriented flight training (LOFT) and may be encountered within a sequence of other events (FAA, 2011a).

However, in informal interviews pilots report that their TCAS training is separated from the realistic complex operational environment. Of note, pilots have reported that the traffic events in their simulator training are predictable and unrealistic. Also, ATC communications are not incorporated in these flights except perhaps by the flight instructor reading out an air traffic instruction.

In 2011, reports submitted to NASA's Aviation Safety Reporting System (ASRS) were studied for descriptions of pilots' interactions with TCAS. Several pilots reported responding to an advisory in a manner that does not mirror FAA guidance, such as performing a horizontal maneuver even though TCAS only delivers vertical advisories. "As the Pilot Flying, the First Officer appropriately initiated a descending left-hand turn away from target per the aural and visual guidance from the TCAS" ACN: 802766, 2008 (Coso, Fleming, & Pritchett, 2011; NASA, 2009)

Likewise, some pilots reported confusion to why TCAS was alerting them or what maneuver TCAS was advising them to perform. "Descending into an airplane that is clearly descending? TCAS software clearly did not give appropriate guidance, nor did it self-correct when the initial guidance was so clearly wrong" ACN: 854982, 2009 (NASA, 2009). Other pilots commented on lack of TCAS training for their particular TCAS unit:

> It was at this point that I attempted to take the TCAS out of the 'traffic' position to the 'TA' to avoid what I considered to be a certain RA in this situation. (Please understand that I just got out of school on this thing in December and there was NO training in regard to differences between the behavior of TCAS in our Airbus fleet and the Boeing fleet.) I flipped the switch from 'traffic' to 'TA/RA' without looking at the switch as I was looking at the traffic.
>
> ACN:820092, 2009 (NASA, 2009)

Reviews of pilots' responses to TCAS have also shown instances of pilots not understanding TCAS (Coso, Fleming, & Pritchett, 2011; Mellone, 1993). One pilot reported flying into the red "do not climb" area displayed on the VSI:

> My FO and I pulled out our sys manual for the CRJ and looked up the different possible indications of the TCAS system and reviewed what had just transpired. We decided that we had both incorrectly interpreted the red 'above' target on the VSI and responded improperly. We further reviewed the procedures, agreeing that a person should fly 'away' from the red VSI indication, if instructed via RA.

However, in our case, we simply should have maintained our current assigned alt of 7000 ft and monitored, as instructed by the TCAS voice.

ACN:785761, 2008 (NASA, 2009)

The TCAS Operational Performance Assessment (TOPA) program found similar pilot responses to TCAS advisories. TOPA monitored the occurrence of TCAS RA's in the terminal area of eight major airports and examined pilot compliance to Climb and Descend advisories. Compliance (yes/no) for a given aircraft was determined if the aircraft achieved a maximum vertical rate greater than zero feet per minute (level) in the appropriate direction. Even this generous measure found pilots only complied in 41% of the observed Climb RA cases and 59% of the observed Descend RA cases (Olson & Olszta, 2010; Olszta & Olson, 2011).

One important concern not included in current (ground or flight) training requirements is overly aggressive pilot responses (Eurocontrol, 2011). Maneuvers in response to corrective TCAS RAs should be initiated with an acceleration (i.e. pull-up or push-down) of 0.25 g to then track the commanded vertical speed (FAA, 2011b). Responding to a corrective TCAS RA should typically cause an altitude deviation of no more than 300 to 500 feet with vertical speeds that are not excessive (FAA, 2011a). However, instances of overly aggressive responses to TCAS RA's have resulted in injuries to crew members and passengers as well as disruptions in air traffic operations. One example of this is the response of Far Eastern Transport Flight EF306 in 2009 to a Descend RA. In this instance, the aircraft was maneuvered into a dive which at one point exceeded 12,000 feet per minute, resulting in the injury of twenty crew and passengers (Aviation Safety Council, 2007). Subsequently, a bulletin released by Eurocontrol in

2011 discussed the need for training objectives intended to prevent excessive responses (Eurocontrol, 2011).

Similarly, a human-in-the-loop study conducted by the Cognitive Engineering Center at Georgia Tech found pilots averaging an altitude deviation of approximately 750 feet for Climb RAs and approximately 1,200 feet for Descend and Crossing Descend RAs (Pritchett et al, 2012b). Likewise, the TCAS Operational Performance Assessment (TOPA) program at MIT Lincoln Labs found similar pilot responses to TCAS advisories. TOPA monitored the occurrence of TCAS RA's in the terminal area of eight major airports and found a maximum altitude deviation of 1,400 feet (Olszta et al, 2011). Altitude deviations of this magnitude could potentially disrupt air traffic operations, as another flight path may be located 1000 feet above or below the pilot's cleared path.

Recently, the FAA released a Safety Alert for Operators. The alert called for all operators with TCAS II to revise their training programs or goals and was released due to data indicating pilots having lower compliance to RA's incurred at lower altitudes, often caused by VFR traffic (Flight Standards Service, 2011).

2.2 Training Program Design

While FAA regulatory material comments on the general goals and requirements of TCAS training programs, literature on training program design in general (including but not necessarily limited to aviation related training programs) provides a broad outlook of the components of an effective training program. Specifically, training programs should highlight four phases of the learning processes: attention, retention, production, and motivation (Bandura, 1986; Mauro & Barshi, 2003; Rosen et al., 2010).

Attention: Attention is the stage whereby learners actively process the information they observe (Rosen, et al., 2010). The attentive state is gained through loud, bright, colorful, or changing features intended to engage the learner. Features that are important to the lesson should be highlighted to keep the student's focus on the relevant material (Mauro & Barshi, 2003; Rosen, et al., 2010).

Retention: Storage of the information takes place in the retention stage, where observed material is encoded into symbolic cues that are then stored in a learner's memory. Retention can be increased by structuring material in a clear and organized manner. The training program can be organized by teaching material in groupings and contexts as they would occur in realistic environments (Mauro & Barshi, 2003; Rosen, et al., 2010).

Production: During the production stage, the information gathered and stored in the previous phases is used to guide an explicit action or behavior (Rosen, et al., 2010). Useful training methods allow learned material to be recalled when needed in the operational context (Mauro & Barshi, 2003).

Motivation: Finally, motivation is the considered as the strengthened likelihood of using an observed (or performed) behavior for future task performance (Mauro & Barshi, 2003). Training should motivate a student to use the learned knowledge accurately and appropriately in future situations. Motivation can also affect the other three stages of learning by driving a student's willingness to be actively engaged in the training material as it is taught (Rosen, et al., 2010).

Not only must a training program highlight these learning processes: a designer must also identify the desired level of performance. Figure 2, copied from Rasmussen (1983), illustrates these three levels, which are described as skill-based, rule-based, and knowledge-based behaviors.

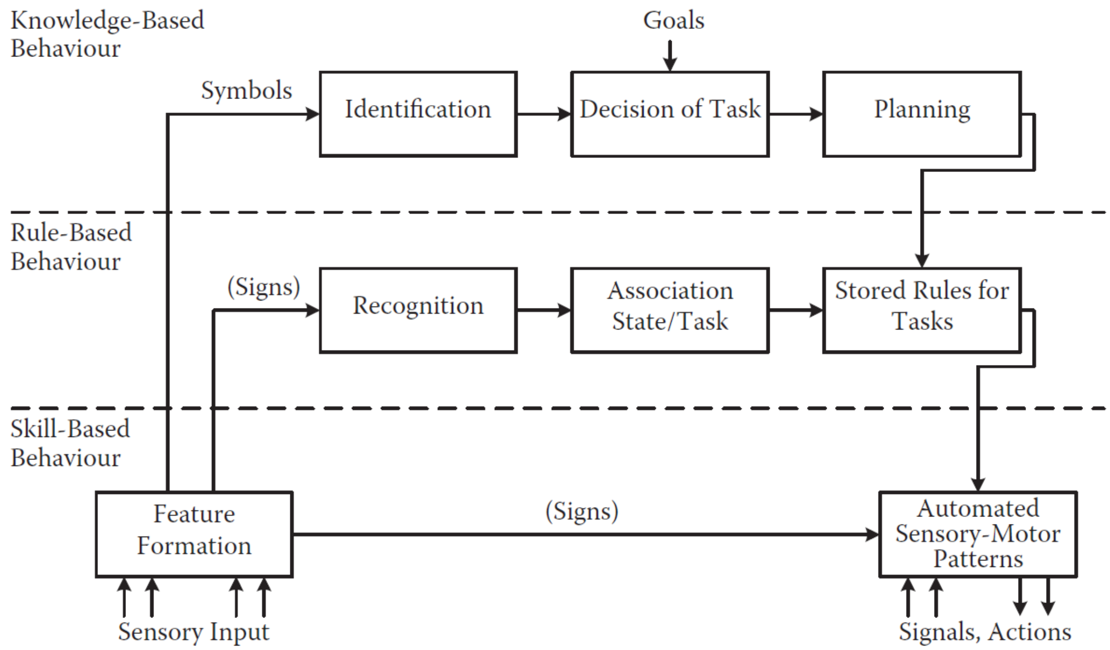


FIGURE 2. Skill-based, rule-based, and knowledge-based behavior (copied from Rasmussen, 1983)

Skill-based behavior represents human performance that does not need conscious control. The human can sense a sign and automatically perform the task based on a simple feedback control. This behavior optimizes human motor skills and needs feedback relative to acceptable performance boundaries (Rasmussen, 1983; Rasmussen, 1989). In response to TCAS advisories, skill-based behavior is exhibited by the pilot’s automatic motor function response to the advisory- disconnecting the autopilot, maneuvering the aircraft manually, and continuously tracking a commanded vertical rate.

Rule-based behavior denotes recognition of a sign or cue and the subsequent performance of a stored rule or procedure. The performance of tasks using rule-based behaviors are goal based, and the rules for task completion are formulated from previous occasions experienced in training or real operations (Rasmussen, 1983; Rasmussen, 1989). For example, pilot understanding of how to achieve TCAS compliance can be supported by training on the rules for following an advisory. Compliance to a TCAS advisory requires the pilot to recognize an aural (and/or visual) cue and then follow a depicted maneuver while also coordinating with ATC. However, the rules pilots have mentally established may or may not align with the actual standards for compliance

Knowledge-based behavior corresponds to situations when higher level cognitive skills are used, and decisions are made by incorporating an analysis of the environment and symbols contained within. Frequently, tasks performed using knowledge-based behaviors take longer to execute, as the human is actively searching for information in novel situations (Rasmussen 1983). Knowledge-based behaviors are exhibited in pilot recognition of the possible traffic event and in their actions before the TCAS advisory. For example, sometimes pilots may be able to predict and anticipate a TCAS event based on information gathered in the moments leading up to the advisory. These expectations may impact compliance (or non-compliance) to the subsequent TCAS advisory.

Appropriate training for all three performance levels should utilize some basic knowledge of effective training(Ivergard & Hunt, 2009. For initial knowledge training followed by task engagement, the complement of two training techniques can be used. Specifically, the integration of *Demonstration Based Training* (DBT) and *Event Based*

Training (EBT) methods aligns with current training division into ground based and flight based training segments.

DBT is a method of training through which the learner acquires knowledge, skills, and attitudes by through various instructional features, including both passive guidance and support as well as observation of demonstrations of task performance (Rosen, et al., 2010; Petrosini et al., 2003; Shlechter & Anthony, 1996). Through imitation, a learner is able to develop the competencies required to reproduce the observed action (Petrosini et al., 2003). While a demonstrated activity often pertains to the specific behavior required to complete a particular task, the training program should also present the broader rules or processes associated with task performance (Shlechter & Anthony, 1996).

Instructional features are used in DBT to provide the learner with supplementary information or activities paired with task performance. Instructional features may include a narrative detailing the procedures for task completion, note-taking exercises requiring learners to be actively engaged, or completion of a short quiz associated to the observed material (Rosen, et al., 2010). Instructional features used in DBT provide the learner with a variety of activities that are inherently embedded with learning opportunities. These instructional features are included before, during, or after the demonstration and are intended to increase attention, production, motivation, and retention (Rosen et al 2010).

Instructional features included as preparatory activities give the learner some introductory knowledge. Concurrent activities are tasks the learner performs while viewing a demonstration and include activities such as note taking. Retrospective tasks are given after the demonstration and may consist of a guided discussion. Finally, prospective activities may incorporate a context or application that wasn’t originally validated in a demonstration and are given after demonstration of task completion. Instructional features categorized as passive guidance or support include additional information relevant to the training objectives and subsequent demonstration of task performance (Rosen et al 2010).

While a first training phase with DBT requires an individual to observe the performance of a task, a second phase with EBT require the learner to perform the task themselves (Rosen, et al., 2010, Fowlkes, Dwyer, Oser, & Salas, 1998). Events introduced in training exercises provide learners with real-world training experience in relevant situations (Fowlkes, Dwyer, Oser, & Salas, 1998).

Design of EBT programs starts with the specification of training objectives and critical tasks. Designers should also consider the context of, and standards for, satisfactory task completion, including desired competencies. After a task is completed, feedback of learner performance should be provided relative to identified standards (Dwyer, Oser, Salas, & Fowlkes, 1999).

EBT asks learners to recall information observed in the DBT phase and to convert the stored information into an overt action. Furthermore, motivation is fostered by the act of practicing a behavior and providing feedback about their task performance (Mauro & Barshi, 2003; Rosen, et al., 2010). Motivatation to use stored knowledge in realistic situations is increased in the EBT phase by asking learners to apply the knowledge, skills, and attitudes.

The complement of DBT and EBT techniques provides the pilot with both knowledge training and engagement in the task. In TCAS training, the outcome of the program would produce rule-based behavior after recognition of a sign (the TA or RA) as well as promote knowledge-based behaviors in those moments preceding the events and skill-based behaviors through the execution of performance (Rosen et al., 2010; Rasmussen 1989).

CHAPTER 3

DESIGN OF AN EFFECTIVE TRAINING PROGRAM

3.1 Overview

The structure of this training program is organized in the same manner as called for in FAA training guidelines with a division between ground-based (e.g. classroom or computer training) and flight training (e.g. simulator training). Integrating Demonstration-Based Training (DBT) for the ground-based training and Event-Based Training (EBT) for the flight training provides a clear division in the apparatus used for employing the training program, with half of the training being completed using a computer-based program and the other half completed using an aircraft simulator under the supervision of an instructor. An added time constraint limited the combined program length to less than 50 minutes, with each segment lasting approximately 25 minutes. This reflects the duration of time available in actual training programs to cover any one aircraft system.

The training objectives for this TCAS training program follow FAA mandated training standards and also address common TCAS misunderstandings. They include general TCAS knowledge, TCAS advisory logic, the mental rules that should be invoked when responding to TCAS advisories, and the signs, signals, and symbols that the pilot can reference about the traffic to properly interact with TCAS, as detailed in the next section.

3.2 Training Objectives

In nominal flight operations both the pilot flying and pilot not flying can monitor and reason about the nearby airspace. Monitoring the nearby airspace is accomplished by a visual scan out the cockpit windows as well as by observing the TSD. The crew also monitors radio communications, both for those communications made directly to the pilot by ATC and for the background chatter, or party-line information, created by communications between other pilots and ATC. Knowledge-based reasoning about these various sources of traffic information can lead to a pilot's prediction of loss of separation with nearby traffic. When a loss in separation is predicted, the pilot may contact ATC for information about the traffic.

If the situation progresses, TCAS advisories demand automatic, i.e. skill-based behaviors, as well as rule-based behaviors. After a TA the pilot is instructed to (if weather permits) scan the nearby horizon for impending traffic but is not authorized to maneuver away from their ATC clearance. Upon an RA, FAA AC155-B mandates that a pilot will comply with the TCAS vertical maneuver even if it maneuvers away from their ATC clearance, except if the pilot believes following the RA would endanger the safety of the flight (FAA, 2011a; FAA 2003). Preventive RA's do not require an immediate change in vertical rate and the pilot does not need to disengage the autopilot. For corrective RA's the pilot-flying is expected to disconnect the autopilot and maneuver the aircraft's vertical speed outside of the indicated "Do Not Fly" or red zone on the VSI. Alternate depictions of RA guidance require the pilot to pitch the nose outside a trapezoid displayed on the PFD. Thus, the RA transfers authority such that the pilot flying can maneuver counter to air traffic instructions.

As shown in Figure 3, TCAS assumes the pilot will initiate a response within five seconds of receipt of a corrective RA and will perform a 0.25 g pull-up (or push-over) to achieve the appropriate vertical rate. The advisory system logic then assumes 2.5 seconds delay and 0.25 g maneuvers for pilot response to any subsequent strengthening or weakening in the commanded vertical speed. No horizontal maneuvers are necessary for compliance to an advisory; indeed, horizontal maneuvers may limit the pilot's ability to meet the required vertical rate. After clear of conflict is annunciated by TCAS, the pilot is expected to return to the originally cleared flight path.

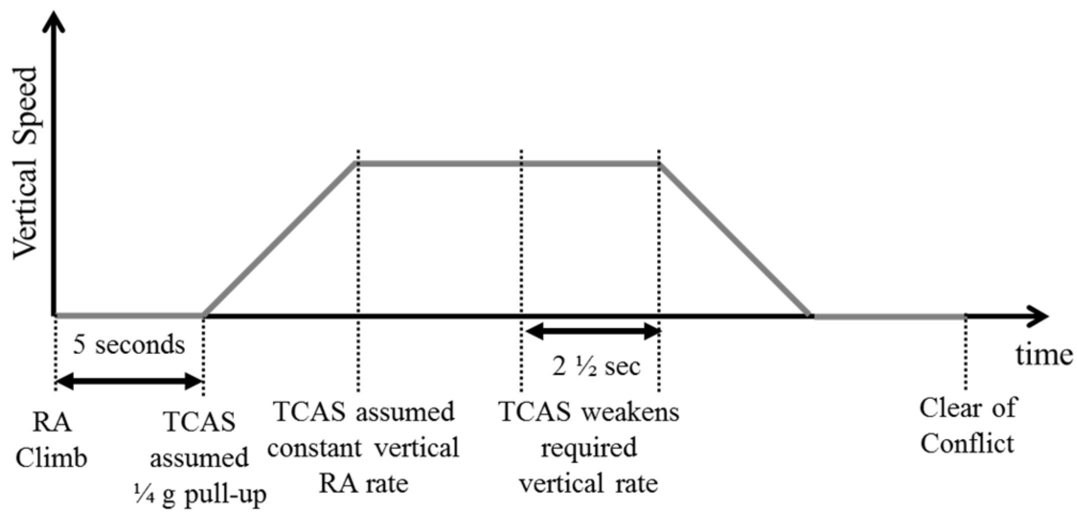


FIGURE 3. TCAS assumed pilot response to advisories. For compliance, pilots are to fly at a vertical rate higher than that directed by TCAS.

This general representation of the expected pilot response requires several specific behaviors. The following sub-sections detail these behaviors, organized around whether they are skill-based, rule-based, or knowledge-based, and identifies specific training objectives for each. The training objectives used for design of the training program are listed in Table 1.

TABLE 1. Training objectives used in training program design

Training Objective	Details	Corresponding Measures
Identify information available prior to an advisory	During nominal operations for approaching traffic, pilots should be able to recognize and monitor the TSD, ATC traffic callouts, ATC party-line information, and out-the-window visual contact	Communications with ATC and co-pilot in training event; Questionnaire responses
Demonstrate an understanding of TCAS logic and assumptions made to generate advisories	Knowledge includes: the information used to generate advisories; TCAS assumes pilot will follow RA within five seconds; TCAS assumes pilot will perform a vertical avoidance maneuver in response to the RA (if needed)	Mid-training quiz; Compliance; Time Pilot First Achieved Compliance; Autopilot Disconnect Time; Questionnaire responses
Distinguish various types of RA's	The various types of RA's taught include: Corrective; Crossing; Reversal; Strengthening; Weakening; Preventive	Compliance; Questionnaire responses
Interpret the commanded RA vertical maneuver	Two types of vertical maneuver guidance depictions are described: VSI- Maintain a vertical speed greater than the red tape; Attitude Indicator- Keep nose outside the trapezoid	Mid-training quiz; Compliance; Time Pilot First Achieved Compliance; Questionnaire responses
Apply rules for compliance when responding to advisories	The rules for compliance include: initiation of vertical avoidance maneuver in the advised direction within five seconds of RA; maintain appropriate rate until "Clear of Conflict"; if ATC gives instructions which conflict with TCAS guidance, pilot must follow RA maneuver	Mid-training quiz; Compliance; Time Pilot First Achieved Compliance; Autopilot Disconnect Time; Questionnaire responses
Distinguish potential negative impacts of an excessive response to an RA	Potential consequences of over-responding to an RA, and provides rules for mitigating excessive responses, such as reducing the aircraft's vertical rate with a weakening RA	Mid-training quiz; Aggressive response features; Questionnaire responses
Apply appropriate rules for response after Clear of Conflict	After Clear of Conflict, pilots should reengage autopilot (if disengaged) and return to their clearance	Post COC Response; ATC Interaction; Questionnaire responses
Identify when it is necessary to communicate with ATC	Pilots should contact ATC when the pilot manuevers from their clearance and/or ATC gives instructions that conflicts with TCAS guidance	Mid-training quiz; ATC Interaction; Questionnaire responses

3.2.1 Skill-Based Behaviors

Skill-based behaviors are exhibited by the pilot's automatic response to TCAS advisories and contribute to the training objectives in Table 1. The sign alerting the pilot to the advisory includes an aural command such as "Climb," "Descend," or "Maintain Vertical Speed." The visual signal for the required vertical maneuver is located on the aircraft's VSI, which the pilot must track by disconnecting the autopilot and controlling the aircraft's pitch (Figure 4).

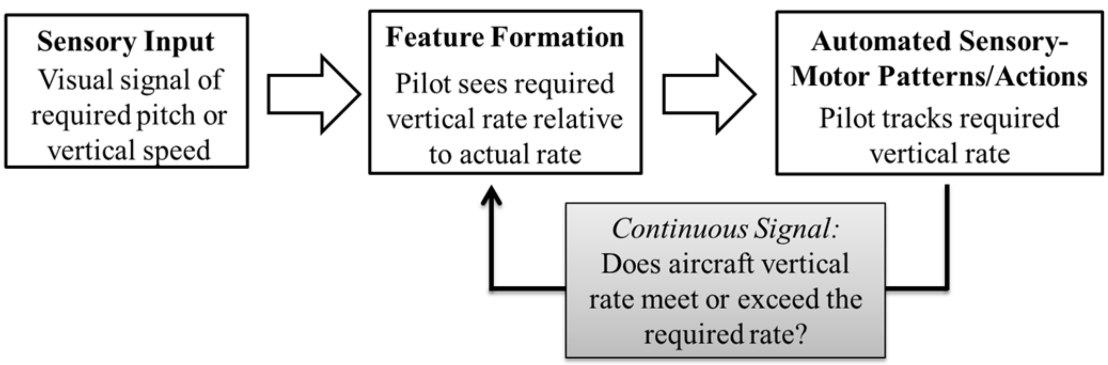


FIGURE 4. Skill-based behavior of a pilot following a TCAS RA

3.2.2 Rule-Based Behaviors

Rule-based behaviors contribute to the training objectives in Table 1 concerning pilot understanding and application of the rules associated with compliance to TCAS advisories and then with their response after Clear of Conflict. The signs provided by TCAS include the aural annunciation as well as the visual TCAS indicators on the TSD and VSI. In addition, ATC may provide signs before and during the advisory, such as air traffic instructions which may conflict with the TCAS RA (Figure 5).

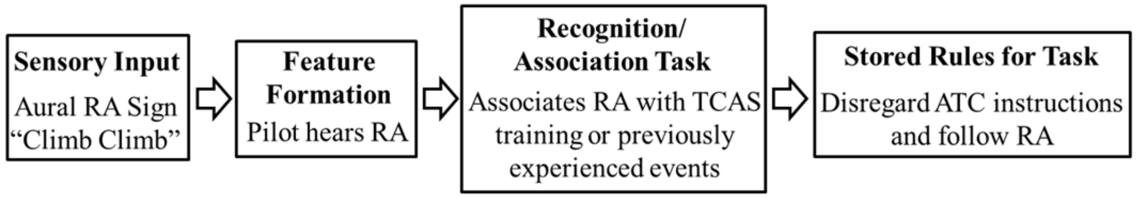


FIGURE 5. Rule-based behavioral aspects of a pilot following a TCAS RA

3.2.3 Knowledge-Based Behaviors

As discussed in the literature review, knowledge-based behaviors correspond to situations when higher cognitive skills are used and decisions are made by incorporating an analysis of the environment with the overall goals of the task. Knolwedge-based behavior may contribute to all the training objectives in Table 1, particularly the objectives that are not focused on rules for compliance and response after Clear of Conflict. Advisories given by TCAS not only encompass aural and visual signs, but also symbols contained in the environment that may be recognized by the pilot at any time. A pilot may be able to recognize a potential traffic hazard by monitoring the TSD, listening to ATC communications to other traffic (also known as party-line information), visually scanning out the window and having knowledge of airspace structures. Training for TCAS can prepare pilots further by teaching them to recognize the different factors that may lead to an event.

Another contributor to knowledge-based behaviors is pilot understanding about why TCAS is generating an advisory. Through this understanding, a pilot's trust of the advisory system may be heightened if TCAS can be perceived as more predictable and dependable.

3.3 Demonstration-Based Training

The first training module applies DBT techniques. DBT is implemented using a computer-based program designed in Microsoft Powerpoint and lasts approximately 25 minutes and Table 2 summarizes the sequence of the DBT content and activities. Appendix A includes the entire set of slides comprising the DBT. The material was organized such that introductory material related to TCAS was taught first, and more difficult topics, such as advisory logic and pilot-ATC interaction, were taught as the lesson developed. DBT uses short quizzes throughout to reiterate information previously taught. Pilots are provided with links back to the relevant material if a mid-training quiz question is answered incorrectly.

TABLE 2. Sequence of DBT content and activities

1. Introduction to TCAS
Discusses the purpose of and need for TCAS Presents basic terminology, and advisory system background to the pilot
2. Overview of components of the Traffic Situation Display
Introduces the pilot to the symbology used for displaying traffic
Mid Training Quiz 1-- TSD Symbology
A TSD is displayed and asked to be interpreted by the pilot
3. Detailed overview of the TA
Discusses the hazard level indicated by a TA (time to closest point of approach) Gives instructions to not perform an avoidance maneuver based solely on the traffic advisory
4. Detailed overview of the RA
Discusses the hazard level indicated by a RA (time to closest point of approach) Establishes clear rules for following RA guidance Adds to knowledge-based understanding of TCAS Provides information related to how TCAS selects specific advisories Presents assumptions made by TCAS advisory logic Presents the various types of RA's (including an aural recording of the alert)
Mid Training Quiz 2-- TCAS advising logic
What information can TCAS use to generate alerts? TCAS advisory logic assumes…
5. Displaying the RA
The various methods of displaying RA guidance are reviewed
6. Excessive response to RA’s
Discusses the potential consequences for over-responding to an RA Provides rules for mitigating excessive responses
7. ATC interaction
Discusses responsibility of pilot to report to ATC if clearance is violated Provides rule for following TCAS guidance when it conflicts with ATC instructions
8. Response after Clear of Conflict
Provides rules for reestablishing cleared path after Clear of Conflict
Mid-Training Quiz 3 -- Response to an Advisory
To comply with a TCAS RA (active or crossing types) Maneuvering at the displayed rate generally causes an altitude deviation of no more than __ Maneuvering well above the advised vertical speed given by TCAS After “Clear of Conflict” is annunciated, I should
Demonstration of the progression of a TCAS advisory
Demonstration includes the signs/signals/symbols available to the pilot

A final review of all the material taught by the DBT segment is achieved through a comprehensive demonstration. The demonstration walks the pilot through the evolution of a TCAS advisory and highlights the information available providing signs, signals, and symbols to the pilot. These signs and symbols are shown in Table 3.

TABLE 3. Rules for compliance to TCAS advisories (in italics) as well as the signs, signals, and symbols available to the pilot.

Before Advisory	
Monitor symbols during nominal operations for approaching traffic	
TSD	Monitor for traffic for possible converging course or traffic symbology indicating "proximate traffic"
Out-the-window	Monitor nearby traffic for possible converging course
ATC	Monitor party-line for nearby traffic or future conflicts. ATC may provide traffic callout.
TA	
Look out window and attempt to establish visual contact	
Aural "Traffic Traffic"	Directs attention to event
TSD	Monitor the approaching traffic's course for continued convergence. Traffic is noted by an amber circle.
RA	
Disconnect autopilot (if required for compliance)	
Maneuver in advised vertical direction	
Maintain appropriate rate until "Clear of Conflict"	
Aural Command	Directs attention to event and cues vertical sense required for compliance
TSD	Monitor the traffic's course for continued convergence. Traffic is noted by a red square.
Red Zone on VSI (or) Trapezoid of PFD	Track required vertical rate relative to actual rate
After Advisory	
Reengage Autopilot	
Return to cleared path (if maneuvered off cleared path)	
Contact ATC and inform of response to advisory (if maneuvered from clearance)	
Aural "Clear of Conflict"	Signifies conflict resolution

Compared to pilots' tasks in everyday-nominal flight operations, TCAS advisories are rarely experienced. However, an immediate and appropriate rule-based response to a TCAS RA is critical for maintaining the safety of the flight. In those infrequent moments when a TCAS RA is given, procedural knowledge describing how the pilot should respond must be quickly transferred from long-term memory to working memory.

Rule-based behaviors are promoted through procedural knowledge constructs using succinct, bullet pointed rules for following a TCAS advisory, e.g. "IF RA is given, THEN follow the guidance" (Anderson, 1983). One example of this is the inclusion of rules for reestablishing the previously cleared path after "Clear of Conflict" (as seen in Figure 6)

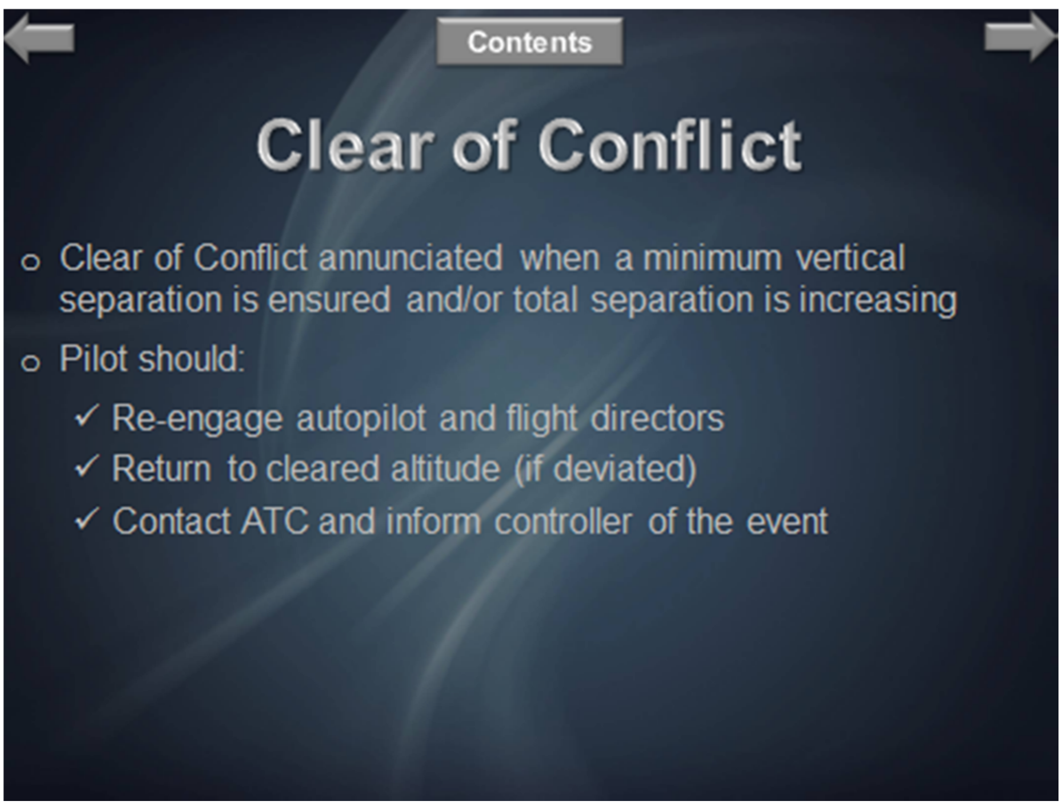


FIGURE 6. Rules pilot response clearance after Clear of Conflict as copied from the TCAS training program

To promote retention, buttons, sounds, images, and animations are incorporated in the DBT segment to engage the pilot in the learning process and to fully portray the important signs and symbols that correspond to the intended rule-based behaviors. In addition, the inclusion of interactive demonstrations allows the pilot to develop the competencies required to for quick retrieval and reproduction of the required rules (Anderson, 1983).

By presenting specific rules for compliance, the training program promotes complying with RA guidance even in the presence of conflicting ATC instructions. Likewise, injury to passengers and crew as well as severe disruption to air traffic operations has been reported due to aggressive responses to RA's. Through a directed lesson, pilots learn about the potential consequences of excessive RA responses and the need for compliance without excessive maneuvering. The presented mental rule for internalization includes mitigating excessive responses to the initial RA while also reducing the aircraft's vertical rate as the RA directs a weakening rate.

Knowledge-based behaviors are also supported through an initial overview of TCAS advisory logic and various situations that may cause a TCAS advisory. These explanations are accompanied by demonstrations of potential scenarios each resulting in a specific advisory. For example, Figure 7 shows the training program's presentation of the logic for crossing RA's.

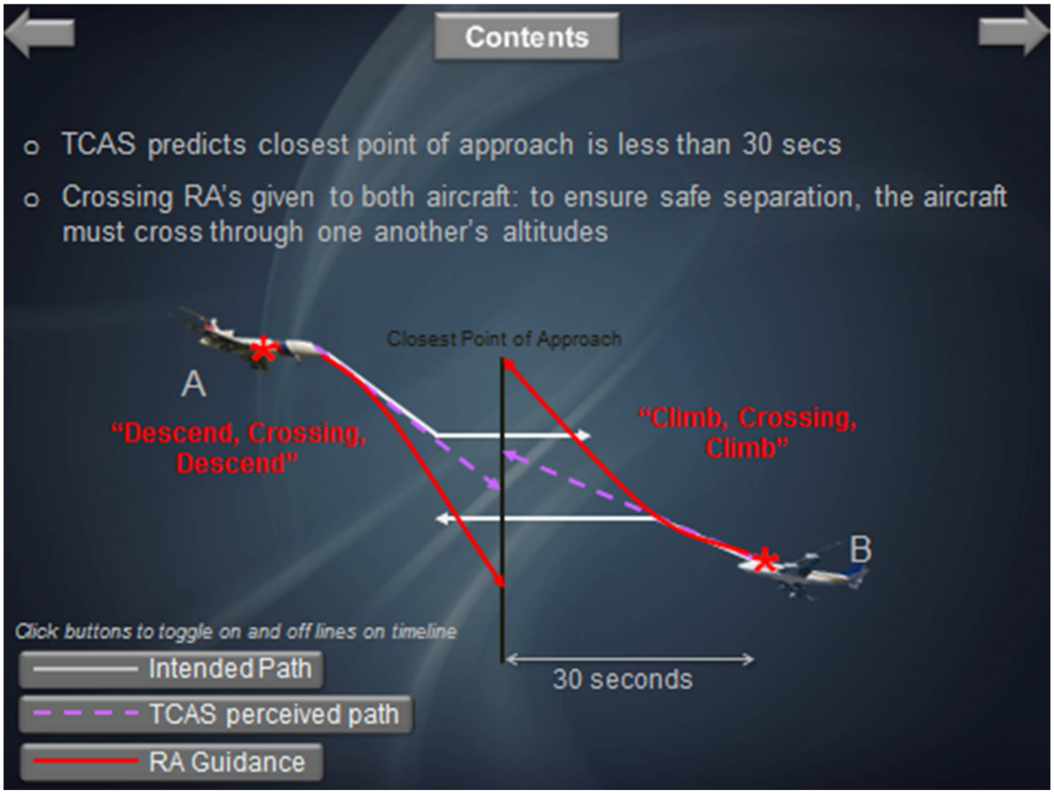


FIGURE 7. Guidance provided by the TCAS training program on the logic behind crossing RA's.

In summary, Table 4 details the contributions of the DBT training phase towards the overall training objectives given earlier in Table 1. The specific rule-based and knowledge-based behaviors associated with each are identified.

TABLE 4. DBT's contribution to the training objectives with assoviated rule-based and knowledge-based behaviors

Training Objective	Details
Identify information available prior to an advisory	Knowledge-based behavior enhanced through discussion and demonstration of the signs, signals, and symbols available to the pilot
Demonstrate an understanding of TCAS logic and assumptions made to generate advisories	Knowledge-based behavior promoted by presenting the underlying logic of advisories
Distinguish types of RA's	Knowledge-based understanding gained through discussion of the different types of RA's and underlying logic
Interpret the commanded RA vertical maneuver	Rule-based behavior promoted through discussion of rules for following the depicted vertical maneuver guidance
Apply rules for compliance when responding to advisories	Rule-based behavior promoted through discussion of rules for complying to an RA
Distinguish potential negative impacts of an excessive response to an RA	Knolwedge-based understanding supported through discussion of potential consequences Rule-based behavior promoted through presentation of rules for mitigating excessive responses
Apply appropriate rules for response after Clear of Conflict	Rule-based behavior promoted through presentation of rules for reestablishing cleared path after Clear of Conflict
Identify when it is necessary to communicate with ATC	Rule-based behavior promoted through presentation of rule for following TCAS guidance when it conflicts with ATC instructions

3.4 Event-Based Training

In typical airline training programs, TCAS flight training is incorporated in the larger scale Line-Oriented Flight Training (LOFT) session where TCAS use and operation is not the main learning objective of the session but instead included as one of many training objectives. Within such training, pilots have reported encountering predicatable TCAS advisories where the traffic flew an unrealistic flight path. Furthermore, if ATC communications were included they were simulated by the instructor.

Lasting approximately twenty-five minutes, the EBT developed here presented events to the pilot that create the requirement to act within a realistic traffic environment. By demanding overt actions, EBT requires the pilot to recall information learned in DBT and apply it in a realistic environment. Since the EBT segment is implemented in an integrated ATC-cockpit simulator, pilots respond to traffic situations in a realistic traffic environment. Use of the appropriate environment reinforces the recognition of signs and signals corresponding to skill-based and rule-based behaviors as well as the application of the symbols and abstractions underlying knowledge-based behaviors. Additionally, the pilots are provided feedback on their performance. Further integration with an ATC simulator provides more realistic TCAS advisories by incorporating convincing target aircraft trajectories as well as accurate ATC communications. The inclusion of realistic environmental context allows for a wider range of training events, resulting in the opportunity to provide the pilot with a broader experience with TCAS in situation where the ATC environment may be important such as when ATC provides conflicting instructions.

EBT focuses on critical tasks that should be performed by the learner, the context of those tasks, and the standard for satisfactory task completion (Dwyer, et al., 1999). For the TCAS EBT, the events are defined by identifying an RA type (e.g. “Climb” or “Descend”), traffic type (e.g. IFR or VFR traffic), and presence of ATC information (e.g. traffic callout or no ATC information). The traffic events were selected to fully exercise the rules as well as the piloting skills required to execute them, as outlined in Table 5.

TABLE 5. Criteria required for successful rule-based and skill-based behavior in EBT events

During RA
Disengagement of autopilot and flight directors *(if needed)*
Maneuver initiated in appropriate vertical direction
Appropriate vertical speed maintained until "Clear of Conflict"
Vertical speed is not excessive
If conflicting ATC guidance given, RA is followed and ATC is informed of maneuver
After Clear of Conflict
Autopilot and flight directors reengaged *(if disengaged)*
ATC notified of TCAS advisory and response *(if clearance was violated)*
Pilot returns to original ATC clearance *(if clearance was violated)*

Knowledge-based behaviors listed in Table 6 could also be applied, but are not necessary for successful task completion. This level of behavior is also reinforced by allowing the pilot to experience multiple types of RA's in varying traffic environments.

TABLE 6. Knowledge-based behaviors applicable in EBT events

Before RA
Identify approaching traffic using the TSD and/or ATC traffic callouts *(if provided)*
Predict and plan for impending TCAS advisory
Discuss impending event with FO
Contact ATC to inquire about nearby traffic
Request new speed/heading/altitude to prevent event *(request would be denied by ATC)*
After Clear of Conflict
Debrief event with FO

Thus, the EBT is intended to allow the pilot to apply rule-based and knowledge-based behaviors covered previously in DBT, re-inforcing their learning. Further, EBT also provides the environment to apply the training objectives for skill-based behaviors. Thus, EBT contributes to all training objectives noted earlier in Table 1.

3.4.1 Simulator

The EBT phase is designed to be completed within a full simulator setup integrating aircraft dynamics with realistic traffic and a realistic air traffic controller. For the purpose of this thesis, the EBT segment was conducted in the Boeing 747-400 Reconfigurable Flight Simulator (RFS) with the Traffic Generation Facility (TGF) air traffic simulator, TCAS emulator, and robust generation of target aircraft trajectories to create the desired traffic event. A diagram of the simulator can be seen in Figure 8.

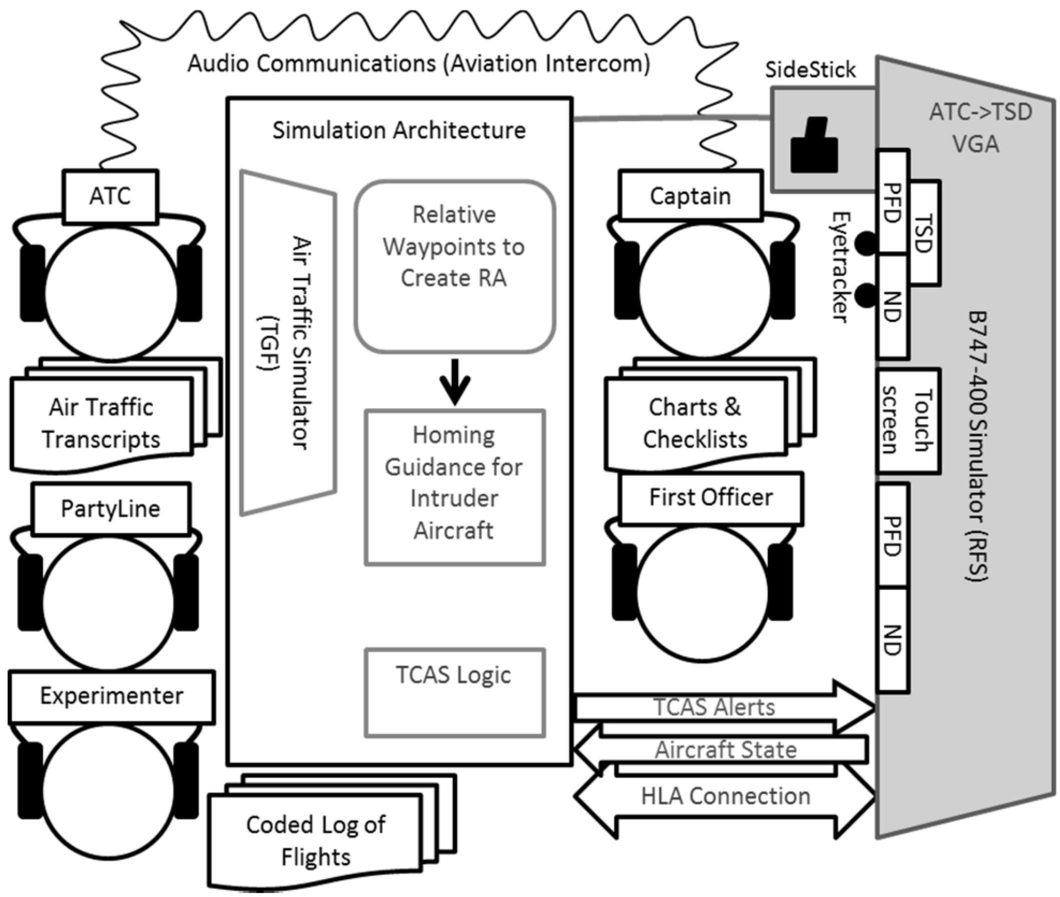


FIGURE 8. Diagram of the Integrated ATC-Cockpit Simulator (Pritchett et al., 2012a)

In each flight, the trainee sat in the left seat, or Captain's seat, and performed the duties of Pilot Flying which are focused on controlling the trajectory of the aircraft, including chosing to maneuver to avoid other aircraft. A staff member who was familiar with the controls of the simulator sat in the right seat posing as the First Officer (FO); in airline operations this could also be a trainee. The FO provided the duties of the "Pilot Not Flying" (or "Pilot Monitoring") in airline operations, which are focused with managing the aircraft systems and interacting with air traffic control. Another staff member posed as the air traffic controller and provided commands to the participant via simulated radio for which the pilot wore a standard flight deck headset. The air traffic controller also communicated with other aircraft in the simulated airspace representing party-line information that the participant was able to overhear. The party-line communication was created by a third staff member.

For the training events, a final staff member in the role of instructor sat behind the Captain and FO. At the end of each training event, the instructor provided feedback which relates to the pilot's performance in preceding event. If needed, constructive criticism can be provided using pre-determined "performance measures." A summary of the various positions, the required duties of each position, and the benefit of including that position is included in Table 7.

TABLE 7. Summary of duties required to operate the integrated simulator

First Officer
Handles duties of Pilot Not Flying, including communications to ATC
Only talks to ATC when instructed to do so by the Pilot Flying during a traffic event
Assistance of FO allows the pilot to focus on responding to traffic events
Adds to realism of event by completing nominal tasks and checklists with pilot
Air Traffic Control Officer
Gives clearances and advisories to the pilot as would be performed in a typical flight
Adds to constructs of environment by giving the pilot an addition source of information
May provide instructions to the pilot which is counter to TCAS guidance
Party-line Information
Adds to realism of event by providing background “chatter” of other traffic in the airspace
Helps promote perception of a "busy" airspace when needed
May provide information to the pilot about an impending situation
Instructor
Gives constructive feedback after conclusion of each event

3.4.2 Training Events

Five training events are presented to the pilot, each designed around a critical task. The context of each training event is varied around the different type of traffic situations that may be encountered (RA type, ATC communications, etc). As the training events progress, advisory context and complexity increases between events. Further, some of the events have a particular importance motivating their inclusion: crossing RAs, conflicting ATC/TCAS guidance (Ladkin, 2004) and an RA caused by VFR traffic (Olszta & Olson, 2011).

Training Event 1. Descend RA

The pilot's first training event is a "Descend" RA with no ATC callouts (Table 8). Acceptable performance is indicated by the pilot disengaging the autopilot and manually achieving the appropriate vertical speed. Also, if the pilot appears to respond with an excessive vertical rate or does not decrease the aircraft's rate appropriately with weakening RA's, his or her performance is critiqued through a verbal debrief with a training instructor. Finally, after Clear of Conflict the pilot is to re-engage the autopilot (and flight directors if turned off). If the pilot deviates from ATC instructions he or she is to return to the previous clearance and contact the controller.

TABLE 8. Training Event 1 – Descend RA

<table>
<tr><th>Training Objective</th><th>Context of Event</th><th>Skill-Based and Rule-Based Behaviors
(Performance Measures)</th><th>Knowledge-Based Behaviors</th><th>Feedback Session</th></tr>
<tr><td>Accurate interpretation of and response to TCAS Descend RA.</td><td>Instrument Meteorological Conditions (daytime, clouds, no winds)
ATC provides no traffic information
Conflict caused by IFR traffic enroute (most likely on departure)
RA maneuvering should not violate ATC instructions
No conflicting ATC or party-line information</td><td>✓ Pilot disengages autopilot and flight directors
✓ Pilot responds to advisory with appropriate vertical speed
✓ Pilot ensures vertical speed is not excessive
o Pilot notifies ATC of response to TCAS advisory as the maneuver is performed
✓ Pilot reengages autopilot and flight directors
✓ Pilot notifies ATC of TCAS advisory and response after clear of conflict
✓ Pilot returns to original clearance (if needed)</td><td>o Identify approaching traffic using the TSD
o Predict and plan for impending TCAS advisory
o Discuss impending event with FO
o Contact ATC to inquire about nearby traffic
o Request new speed/heading/altitude to prevent event (request would be denied by ATC)
o Debrief event with FO</td><td>If the pilot did not meet any particular performance measure, review the correct response in regards to that measure</td></tr>
</table>

✓ indicates pilot must perform this action

o indicates pilot should, but does not have to, perform this action

Training Event 2. Climb RA

The pilot's second training event is a "Climb" RA with ATC callouts for the traffic (Table 9). Acceptable performance is indicated by the pilot disengaging the autopilot and manually achieving the appropriate vertical speed. Also, if the pilot appears to respond with an excessive vertical rate or does not decrease the rate appropriately with weakening RA’s, his or her performance is critiqued through a verbal debrief with a training instructor. Finally, after Clear of Conflict the pilot is to re-engage the autopilot (and flight directors if turned off). If the pilot deviates from ATC instructions he or she is to return to the previous clearance and contact the controller.

TABLE 9. Training Event 2 – Climb RA

Training Objective	**Context of Event**	**Skill-Based and Rule-Based Behaviors** *(Performance Measures)*	**Knowledge-Based Behaviors**	**Feedback Session**
Accurate interpretation of and response to TCAS Climb RA.	Instrument Meteorological Conditions (daytime, clouds, no winds) ATC provides traffic callout Conflict caused by IFR traffic enroute (most likely on departure) No conflicting ATC or party-line information	✓ Pilot disengages autopilot and flight directors ✓ Pilot responds to advisory with appropriate vertical speed ✓ Pilot ensures vertical speed is not excessive o Pilot notifies ATC of response to TCAS advisory as the maneuver is performed ✓ Pilot reengages autopilot and flight directors ✓ Pilot notifies ATC of TCAS advisory and response after clear of conflict ✓ Pilot returns to original clearance (if needed)	o Identify approaching traffic using the TSD and/or ATC Traffic Callout o Predict and plan for impending TCAS advisory o Discuss impending event with FO o Contact ATC to inquire about nearby traffic o Request new speed/heading/altitude to prevent event *(request would be denied by ATC)* o Debrief event with FO	If the pilot did not meet any particular performance measure, review the correct response in regards to that measure

✓ indicates pilot must perform this action

o indicates pilot should, but does not have to, perform this action

Training Event 3. Crossing Descend RA

Previous studies (Pritchett et al, 2012a; Pritchett et al, 2012b; Pritchett et al, 2012c) have shown pilots not understanding or trusting “Crossing” RAs. These RAs ask the pilot to descend or climb through the opposing traffic’s altitude. Therefore, the pilot's third training event is a "Crossing Descend" RA with no ATC callouts (Table 10). An appropriate response to the RA will cause a deviation from ATC instructions.

Acceptable performance is indicated by the pilot disengaging the autopilot and manually achieving the appropriate vertical speed. Also, if the pilot appears to respond with an excessive vertical rate or does not decrease the rate appropriately with weakening RA’s, his or her performance is critiqued through a verbal debrief given by a training instructor. Finally, after Clear of Conflict the pilot is to re-engage the autopilot (and flight directors if turned off). If the pilot deviates from ATC instructions he or she is to return to the previous clearance and contact the controller.

TABLE 10. Training Event 3 – Crossing Descend

Training Objective	**Context of Event**	**Skill-Based and Rule-Based Behaviors** *(Performance Measures)*	**Knowledge-Based Behaviors**	**Feedback Session**
Accurate interpretation of and response to TCAS Crossing Descend RA	Instrument Meteorological Conditions (daytime, clouds, no winds) ATC provides no traffic information Traffic is other commercial type aircraft (most likely on departure) RA maneuvering violates ATC instructions No conflicting ATC or party-line information	✓ Pilot disengages autopilot and flight directors ✓ Pilot responds to advisory with appropriate vertical speed ✓ Pilot ensures vertical speed is not excessive o Pilot notifies ATC of response to TCAS advisory as the maneuver is performed ✓ Pilot reengages autopilot and flight directors ✓ Pilot notifies ATC of TCAS advisory and response after clear of conflict ✓ Pilot returns to original clearance (if needed)	o Identify approaching traffic using the TSD o Predict and plan for impending TCAS advisory o Discuss impending event with FO o Contact ATC to inquire about nearby traffic o Request new speed/heading/altitude to prevent event *(request would be denied by ATC)* o Debrief event with FO	If the pilot did not meet any particular performance measure, review the correct response in regards to that measure

✓ indicates pilot must perform this action

o indicates pilot should, but does not have to, perform this action

Training Event 4. Conflicting ATC Information

In the case of the Uberlingen mid-air collision, the pilot of a Russian chartered flight followed ATC instructions to descend, despite conflicting guidance from TCAS to climb (Table 11). To simulate this, the pilot's fourth training event includes conflicting guidance from ATC and TCAS: ATC instructs the pilot to descend for traffic just before TCAS gives a "Climb" RA.

Acceptable performance is indicated by the pilot complying with the RA through the disengagement of the autopilot and manually achieving the appropriate TCAS directed vertical speed. After initiating a response, the pilot should call ATC and inform the controller of the compliance to TCAS. Also, if the pilot appears to respond with an excessive vertical rate or does not decrease the aircraft’s rate appropriately with weakening RA’s, his or her performance is critiqued through a verbal debrief with a training instructor. Finally, after Clear of Conflict the pilot is to re-engage the autopilot (and flight directors if turned off) and return to their clearance.

TABLE 11. Training Event 4 – Conflicting ATC information results in Climb RA

Training Objective	**Context of Event**	**Skill-Based and Rule-Based Behaviors** *(Performance Measures)*	**Knowledge-Based Behaviors**	**Feedback Session**
Accurate interpretation of and response to TCAS Climb RA. ATC gives conflicting instructiond (to descend) just before Climb RA is given.	Instrument Meteorological Conditions (daytime, clouds, no winds). ATC provides conflicting guidance. Conflict caused by IFR traffic enroute (most likely on departure). RA maneuvering may violate ATC instructions.	✓ Pilot disengages autopilot and flight directors ✓ Pilot responds to advisory with appropriate vertical speed ✓ Pilot ensures vertical speed is not excessive ✓ Pilot notifies ATC of response to TCAS advisory as a maneuver is performed and rejects ATC guidance ✓ Pilot reengages autopilot and flight directors ✓ Pilot notifies ATC of TCAS advisory and response after clear of conflict ✓ Pilot returns to original clearance (if needed)	o Identify approaching traffic using the TSD and/or ATC Traffic Callout o Predict and plan for impending TCAS advisory o Discuss impending event with FO o Contact ATC to inquire about nearby traffic o Request new speed/heading/altitude to prevent event *(request would be denied by ATC)* o Debrief event with FO	If the pilot did not meet any particular performance measure, review the correct response in regards to that measure

✓ indicates pilot must perform this action
o indicates pilot should, but does not have to, perform this action

Training Event 5. RA Caused by VFR Traffic

The final training event consisted of a "Preventive: Do Not Descend" RA caused by VFR traffic passing 500 feet below the pilot's aircraft (Table 12). ATC does give the pilot a callout for the traffic. This event is included because approximately 85% of RA's are believed to be caused by "low-performance General Aviation" aircraft (Olszta & Olson, 2011). Additionally, the RA is triggered even though the aircraft are legally separated by the 500 feet separation between IFR and VFR altitude.

Acceptable performance is indicated by the pilot maintaining a vertical rate above the "red" region on the VSI. Disengagement of the autopilot is not necessary to comply. An appropriate response to the RA will not cause any deviation from ATC instructions, thus reporting of the event to ATC should not be necessary.

TABLE 12. Training Event 5 – Preventive RA caused by VFR traffic

Training Objective	**Context of Event**	**Skill-Based and Rule-Based Behaviors** *(Performance Measures)*	**Knowledge-Based Behaviors**	**Feedback Session**
Accurate interpretation of and response to TCAS Preventive RA, caused by VFR traffic passing 500 feet below	Instrument Meteorological Conditions (daytime, clouds, no winds) ATC provides traffic information Traffic is VFR aircraft Preventive RA (no maneuvering required) No conflicting ATC or party-line information	✓ Pilot does not disengage autopilot and flight directors o Pilot notifies ATC of response to TCAS advisory as the maneuver is performed ✓ Pilot notifies ATC of TCAS advisory and response after clear of conflict	o Identify approaching traffic using the TSD and/or ATC Traffic Callout o Predict and plan for impending TCAS advisory o Discuss impending event with FO o Contact ATC to inquire about nearby traffic o Request new speed/heading/altitude to prevent event *(request would be denied by ATC)* o Debrief event with FO	If the pilot did not meet any particular performance measure, review the correct response in regards to that measure

✓ indicates pilot must perform this action
o indicates pilot should, but does not necessarily have to, perform this action

CHAPTER 4

EVALUATING TRAINING PROGRAM EFFECTIVENESS

4.1 Overview

The efficacy of this training program was carefully evaluated through the duration of the DBT and EBT training phase and at the completion of the training program. Pilot performance is compared to an earlier study conducted with the baseline of current airline pilots. The overall goal of this study was to answer the research question: "Does the training program meet the first objective of this thesis? I.e., does it train pilots to understand TCAS use for collision avoidance in the actual traffic and operational environment?"

4.2 Experiment Design

4.2.1 Participants

A prior study conducted in January 2012 serves as a baseline for comparison with the trained pilots in this study. Both the baseline and the trained pilots had been previously trained for TCAS by their carrier; the term "trained pilots" is used here to denote pilots who completed the modified training program.

Sixteen pilots participated in the baseline study. All of the participants were male, ranging in age from their mid-20's to 59 years old. Eight held the rank of Captain in their airline, seven were ranked as First Officers, and one did not respond to the question. Eight of the sixteen pilots reported having received some form of military training in their aviation career. Eight of the pilots reported being very familiar with the airport used for this study (Dallas- Fort Worth), seven of the pilots had some familiarity with the airport, and one pilot reported having no familiarity.

Eighteen pilots participated in the training study, recruited using a nearly identical recruiting protocol to the baseline study. All of the participants were male, ranging in age from their mid-20's to 59 years old. Four of the participants held the rank of Captain and fourteen were ranked as First Officers. Five reported having some form of prior military training related to aviation. Five pilots reported being very familiar with Dallas-Fort Worth airport, eight had some familiarity, and five reported having no familiarity.

4.2.2 Experiment Apparatus

The study was conducted using the same full simulator setup as used in the EBT which integrated aircraft dynamics with realistic traffic and a realistic air traffic controller, described in Chapter 3. With the integrated simulator, it was possible to vary the amount and type of traffic information available to the pilot through the pre-scripted party-line transcripts and ATC traffic callouts. The perceived traffic density was also varied by selecting scenarios with many (or few) targets displayed on the pilot's TSD as well as by controlling the frequency of party-line information (e.g. 15-30 seconds between party-line callouts indicated a heavy traffic density). Additionally, the target aircraft's trajectory was defined by a series of waypoints established using realistic routes appropriate to the airspace; only once these routes placed it near the ownship did the target aircraft initiate a course creating an RA. Thus, approaching aircraft trajectories were realistic in that they appeared to be flying along a published route.

4.2.3 Experiment Procedure

The procedure comprised of four segments as shown in Table 13. First, the experiment required some administrative activities before the participant could complete the training. Second, pilots completed the modified DBT/EBT training program. Then, third, the pilots were asked to respond to multiple traffic events in the simulator and their behavior was observed. Finally, the pilots were de-briefed.

TABLE 13. Overview of TCAS training program and evaluation

	Activity	Medium	Description
Initial Paperwork (50 minutes)	Informed Consent and Briefing	Printed	Outlines risks of study Outlines the training program, simulator, and study purpose
	Pre-Experiment Questionnaire	Computer Based	Gathers information regarding pilot's previous interactions with TCAS Gathers information regarding pilot attitude toward TCAS (can use an abbreviated version of previous pre-experiment questionnaire)
	Pre-Experiment Quiz	Computer Based	Tests the pilot on their knowledge of TCAS Does not consider the pilot's attitude toward TCAS
	Simulator Training	Simulator Based	Allows the pilot to fly a simulator tutorial flight, with the aid of a first officer acting as Pilot Monitoring
Training Program (50 minutes)	Demonstration Based Training	Computer Based	Presents 1 possible traffic event to the pilot and the different outcomes
	Event Based Training	Simulator Based	Presents traffic events in the simulator that create the requirement to act, followed by a short debriefing Events sequenced to bring in more context and complexity
10 Minute Break			
Evaluating TCAS Interactions Post Training (2 hours)	Flight Scenarios	Simulator Based	Presents traffic events in the simulator that create the requirement to act No feedback is given after each encountered event Ordered using a Latin Squares
	Post Scenario Questionnaires	Hand-written	Gathers information regarding the pilot's perception of the traffic events that had just occurred
De-Brief (15 minutes)	Post-Experiment and Training Questionnaire	Computer Based	Gathers information regarding the pilot's demographics Inquires about the pilot's attitude toward the modified training program as well as their interest in any additional information not included in the training program

Each experimental flight asked the pilots to fly an approach course into Dallas-Fort Worth airport. Each scenario lasted approximately fifteen minutes, generally starting at an altitude of roughly 10,000 to 20,000 feet above mean sea level and ending when the pilot intercepted the final approach course. Two traffic events were created within each scenario. In each experimental flight, similar to the training flights the participant sat in the left seat, or Captain's seat, and performed the duties of Pilot Flying which include controlling the trajectory of the aircraft, monitoring the airspace for traffic, and responding to any traffic events and TCAS advisories. A researcher familiar with the controls of the simulator sat in the right seat posing as the First Officer (FO). The FO provided the duties of the "Pilot Not Flying" (or "Pilot Monitoring") in airline operations, which include managing the aircraft systems and interacting with air traffic control. The FO neither helped with the traffic nor intentionally distracted the pilot from the event. Another researcher posed as the air traffic controller and provided commands to the participant via simulated radio for which the pilot wore a standard flight deck headset. The air traffic controller also communicated with other aircraft in the simulated airspace representing party-line information that the participant was able to overhear. The party-line communication was simulated by a third researcher. For the training events, a final researcher sat behind the Captain and FO and was responsible for monitoring and configuring the simulator. Unlike the training phase when this position acted as an instructor in the experimental phase this researcher did not provide feedback performance. At the end of each experimental scenario, the pilot was asked to complete a post-scenario questionnaire.

4.2.4 Independent Variables and Experimental Design

The pilots experienced twelve traffic events. The run order of the scenarios was varied between pilots using a Latin Square design to compensate for possible run order effects. As shown in Table 14, three fixed variables defined each traffic event: the information provided to the pilot about traffic prior to the TCAS event (i.e., ATC Information), traffic density, and the TCAS advisory created by the target aircraft's trajectory. The ATC information about the traffic was varied at three levels: Callout, Party-line, and Conflicting. For events with a traffic callout the air traffic controller gave a call-out to the pilot about the other aircraft prior to the TCAS event. In events containing party-line information the background radio chatter from other traffic contained relevant information about the target aircraft but may or may not have been recognized by the pilot. In those events with conflicting guidance, the air traffic controller instructed the pilots to "descend for traffic" moments before the pilot received a TCAS advisory to Climb. The traffic density was varied at two levels: light and heavy. The density represented a subjective measure of how congested the airspace appeared on the TSD (many versus few targets on the TSD). Also, ATC communications with other traffic occurred at a much higher rate in heavy traffic density, creating the appearance that the controller was extremely busy. The TCAS advisory type was varied between TA only, Climb RA, Descend RA, Crossing Descend RA, and Preventive. In two of the events VFR traffic triggered the advisory.

TABLE 14. Traffic events and corresponding independent variables

Event	Advisory Type and Traffic Trajectory	Traffic Information	Traffic Density
Training 1	None		
Training 2-Event 1	TA	Party-line	Light
Training 2-Event 2	Descend	Callout	Light
A-Event 1	TA	Callout	Heavy
A-Event 2	Climb	Party-line	Heavy
B-Event 1	Descend	Conflicting	Light
B-Event 2	Crossing	Party-line	Light
C-Event 1	Descend	Party-line	Heavy
C-Event 2	Climb (VFR)	Callout	Heavy
D-Event 1	TA (VFR)	Party-line	Light
D-Event 2	Preventive	Callout	Light
E-Event 1	Crossing	Conflicting	Heavy
E-Event 2	Climb	Callout	Heavy
F-Event 1	Crossing	Callout	Heavy
F-Event 2	TA	Party-line	Heavy

The results from the initial baseline study identified events of interest. For example, the baseline study showed pilots exhibiting little knowledge of the counter-intuitive Crossing Descend RA, with one pilot interpreting the advisory as a system error. This result motivated the inclusion of the same event for direct comparison (Pritchett et al, 2012a; Pritchett et al, 2012b; Pritchett et al, 2012c).

Other events, such as that with VFR traffic, were selected based on results from research monitoring actual pilot response to TCAS advisories. Particularly, the VFR events were included because approximately 85% of RA's are believed to be caused by "low-performance General Aviation aircraft" (Olszta & Olson, 2011). These General Aviation aircraft follow a VFR altitude, triggering an RA even though the aircraft are legally separated.

Each event in the training study was comparable to an event from the baseline study. The events were considered comparable if all three independent variables were the same level as the matched event in the baseline study. For example, Event A2 was a climb RA with party-line information and heavy traffic density; the baseline study also had an event with a climb RA, party-line information, and heavy traffic density that was paired with Event A2 for statistical comparison. However, the traffic events were not paired into flight in the same combinations as in the baseline study because the pilots in this study completed two less flights than the baseline pilots due to time limits.

4.3 Measures of Effectiveness Overview

This section examines pilot behavior prior to completing this training, during the training program, and during the experiment after training. This discussion is organized around analyses of the pilots' skill, rule, and knowledge-based behaviors.

4.3.1 Skill-Based Behaviors

As discussed in the literature review and in Chapter 3, skill-based behaviors encompass recognition of a sign and automatic performance of a physical response tracking a signal. Two dependent variables describe automatic behaviors: the measure Autopilot Disconnect Time After RA Initiation, and pilot tracking of the vertical rate commanded by the RA as captured in the measure Time Pilots First Achieved Compliance After RA Initiation. Figure 9 depicts a graphical example of each measure.

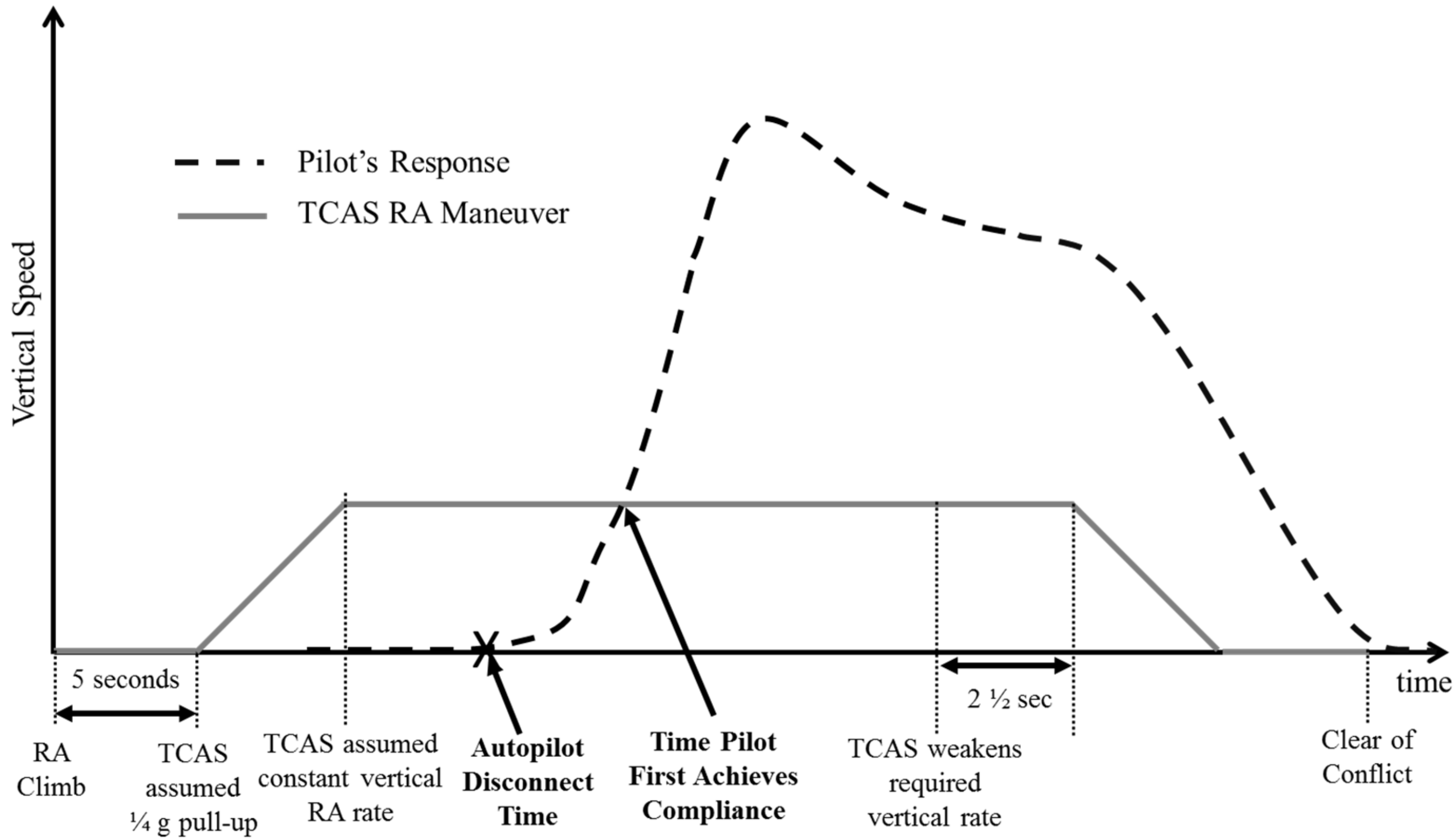


FIGURE 9. TCAS manuever, pilot response, and corresponding measures of skill-based behaviors

The Autopilot Disconnect Time After RA Initiation indicates how quickly the pilot is able to sense the sign for the advisory and perform the skill-based physical response of autopilot disconnect. Follow EBT is predicted to decrease this measure.

The Time Pilots First Achieved Compliance After RA Initiation was measured by the time after the RA when the pilot first meets or exceeds the TCAS directed vertical rate. While responding to an RA, the pilot is continuously tracking the actual aircraft vertical rate compared to the required vertical rate for compliance and making adjustments accordingly. An increase in performance at the skill-based level would improve the pilot's ability to make precise adjustments, and thus training is expected to decrease the time compliance is first achieved.

4.3.2 Rule-Based Behaviors

Rule-based behaviors encompass the recognition of a sign and subsequent initiation of a stored rule. The rule-based behaviors of interest for this study can be examined by measures of aggressive response features, compliance rates, the measure Pilot Response After Clear of Conflict, and the measure Pilot Interaction With ATC.

Aggressive Response

Aggressiveness is quantified and analyzed using four dependent measures: Altitude Deviation Over Duration of RA, the Average Vertical Rate Difference, the Maximum Vertical Rate Difference, and the Maximum Vertical Rate. After completing the training program, measures of aggressive RA responses are expected to decrease. Figure 10 depicts a graphical example of how the vertical rate measures can be viewed;

the measure Altitude Deviation Over Duration of RA is the integral of the pilot's vertical rate.

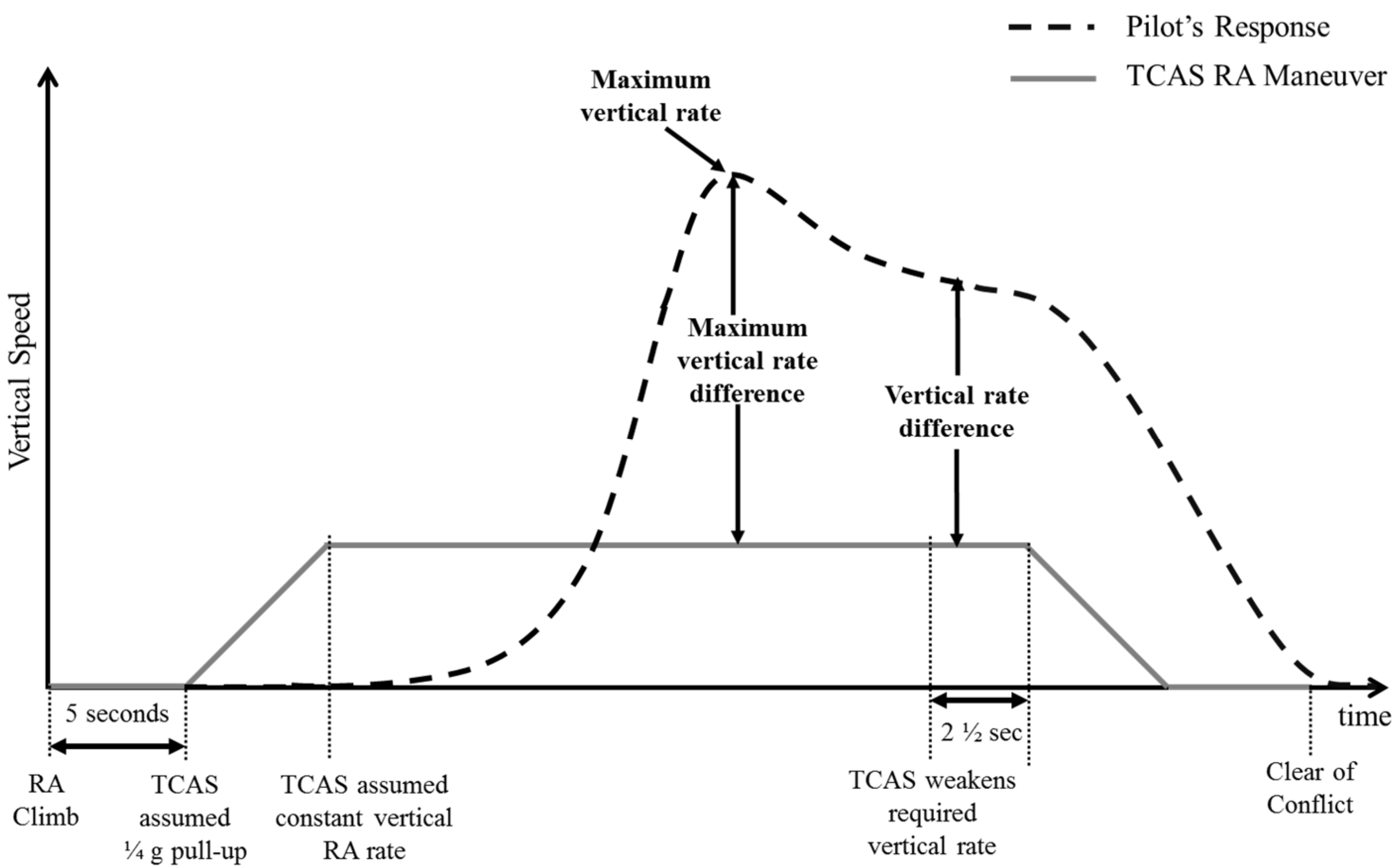


FIGURE 10. TCAS manuever, pilot response, and corresponding measures of aggressiveness

Compliance

Figure 11 depicts a graphical example of how compliance is measured. The measure Percentage Compliance is calculated by dividing the amount of time the pilot was in compliance to the RA by the total duration of the RA. The measure Binary Value for Compliance "Yes" (or 1) if the pilot complied to the RA for the entire advisory duration (i.e., percentage compliance = 100%) and "No" (or 0) if the pilot did not comply to the RA for the entire advisory duration (i.e., percentage compliance < 100%).

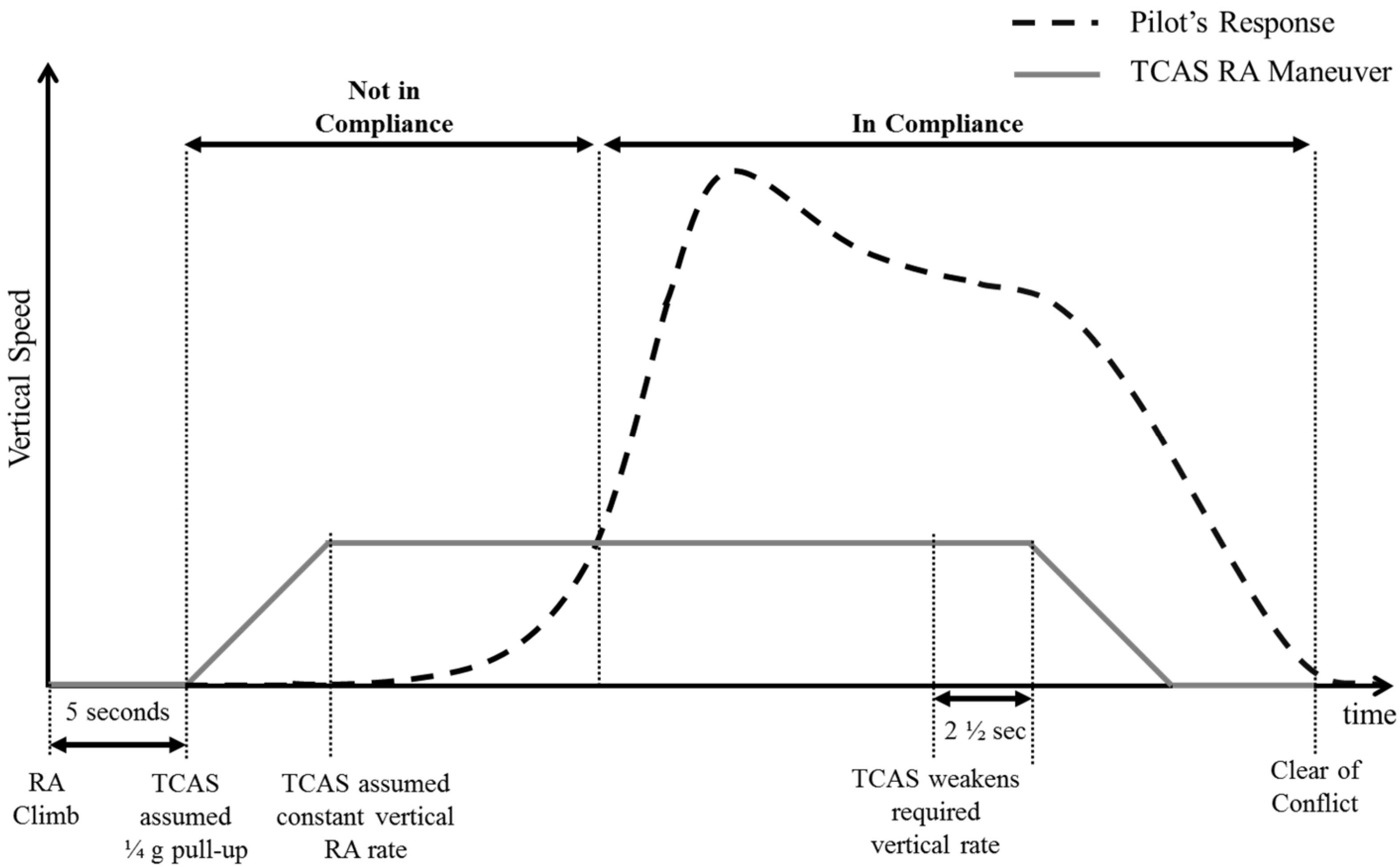


FIGURE 11. TCAS manuever, pilot response, and corresponding assessment of when the pilot is in compliance

Pilot Response After Clear of Conflict

Pilot Response After Clear of Conflict was used to analyze a pilots' reaction after the RA, and if they returned to their cleared path (or not). If the pilot did not immediately return to the cleared path, but instead contacted ATC for further instructions, a "Yes" (or 1) was recorded by the experimenter for a binary measure indicating if the pilot returned to the previously cleared path after response to the RA. Otherwise, if the pilot did return to his cleared path the experimenter recorded a "No" (or 0). Figure 12 has an example of the different ways a pilot can respond after Clear of Conflict.

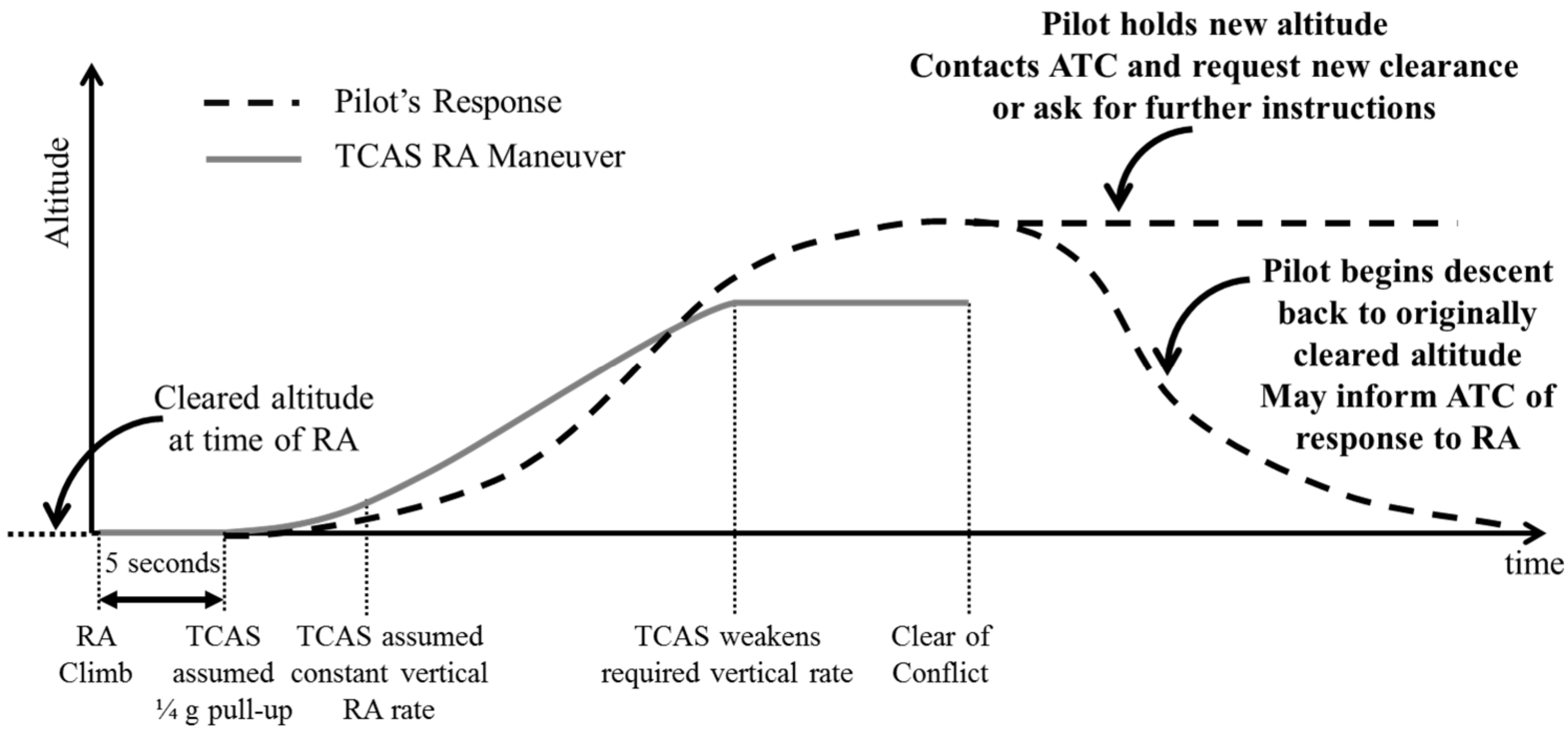


FIGURE 12. TCAS manuever, pilot response, and corresponding measures of response after clear of conflict

Pilot Interaction With ATC

Lastly, the measure Pilot Interaction With ATC can be studied in the context of rule-based behaviors. This is done by examining binary measures related to if the pilot contacted ATC after the TA as well as if the pilot contacted ATC after the RA. Analysis of this interaction (regardless of whether or not the pilot followed ATC instructions) gives insight to whether the pilot has established a consistent rule for communicating with ATC after a TCAS advisory.

4.3.3 Knowledge-Based Behaviors

While the skill-based and rule-based behaviors were primarily observable through observations in the simulator, the knowledge-based behaviors were primarily observable through responses to the questionnaires, providing insight related to pilot trust and understanding of TCAS. Responses to questions about the training program indicate which features the pilot did and did not like about the program and are one indicator of the pilot's use of –and attitude toward– the training program.

While the knowledge-based behaviors are primarily observable through subjective questionnaire responses, ATC interaction may indicate reasoning prior to an event. The measure Pilot Interaction With ATC before receiving any TCAS advisory (TA or RA) shows recognition of a possible event through the information provided by his general environment, while the measure Pilot Interaction With ATC after clear of conflict indicates reflection about the event.

4.4 Data Collection

Several types of data were gathered throughout the study. Pilots completed three questionnaires and pilot understanding of TCAS was assessed via a "Pre-Training TCAS Quiz." Throughout the flights, the simulator logged aircraft state, TCAS advisories, and audio communications. Documents used in the experiment are located Appendix B and each of these data collection methods is described in the following subsections.

4.4.1 Pre-Training TCAS Quiz

Before completing the training program, each pilot was asked to complete a brief "Pre-Training TCAS Quiz." The quiz comprised of 11 multiple choice questions, each related to material taught in the modified training program (Table 15). Several questions were repeated in the "Mid-Training Quizzes" used in DBT. The quiz served as a method for judging the specific knowledge the pilot had prior to completing the training program.

TABLE 15. Pre-Training Quiz question topics

Question Topic
TCAS advising logic
Assumptions made by TCAS
Interpreting the Traffic Situation Display
Actions required for compliance
Aggressiveness
Actions required after post-Clear of Conflict
Conflicting ATC-TCAS guidance

4.4.2 Questionnaires

Unlike the Pre-Training Quiz, the "Pre-Experiment Questionnaire" asked questions focusing on the pilot's subjective perceptions of TCAS. Pilot responses were given on a Likert Scale. Here, the pilot selected whether he Strongly Disagrees/Disagrees/Is Neutral/Agrees/Strongly Agrees with given statements related to TCAS use and operation. The questionnaire also asked about procedures for following an RA as directed by his airline and previous TCAS training (Table 16). After each experimental flight, a one-page "Post Scenario Questionnaire" gathered qualitative information from the pilot about the previous traffic events (Table 17).

TABLE 16. Overview of Pre-Experiment Questionnaire

Pre-Experiment Questionnaire
Likert Scale Questions
TCAS understanding
Trust in TCAS
Past compliance to advisories
Using TCAS for maintaining safe separation
Usefulness of the TSD
Free Response Questions
Current airline's procedures for complying to an RA
Previous TCAS training

TABLE 17. Overview of the Post-Scenario Questionnaire

Post-Scenario Questionnaire
Likert Scale Questions
Agreement with TCAS
TCAS understanding
Trust in TCAS
Ability to predict advisory
Timing of event
Perceived cause of event (ATC or other pilot's actions)
Necessity of advisory
Perceived conflicting guidance from ATC
Free Response Questions
Information sources used to determine actions

The “Post-Experiment Questionnaire” asked the pilot to fill out information about his background as well as his perception of the training program and his perception of the overall experiment (Table 18). This questionnaire started by gathering the pilot’s demographic attributes. Subsequent questions asked the pilot to provide feedback on the training program. Many of the questions were formatted using a Likert Scale and were associated with different aspects of the training program. Also, the pilot was asked to muse about their experience through open ended questions. Finally, pilots had an opportunity to assess the overall experiment: questions asked if the pilot was able to deduce any patterns in the traffic scenarios, if the simulator hindered his true response (and if so how he would have responded in revenue flight) and also if his response would have changed in visual meteorological conditions.

TABLE 18. Overview of the Post-Experiment Questionnaire

Post-Experiment Questionnaire
Demographics
Age
Gender
Rank
Prior military experience
Aircraft experience (type and hours)
Familiarity with airspace
TCAS implementation used on current aircraft
Training Program
Likert Scale Questions
Ease of using CBT/EBT
Ease of understanding CBT/EBT
General attitude (positive/negative) of using CBT/EBT
Integration of CBT/EBT
Training's effect on TCAS knowledge and use
Training's effect on aggressive response features
Trust in TCAS after training
Similarity of program to previous training
Length of program
Free Response Questions
Material learned from training program
General comments
Flight Simulation
Free Response Questions
Observed patterns
Ease of using the simulator
Effect of IFR conditions

4.4.3 Performance in Flight

Recorded Measures

The integrated simulator provided data for the pilot's control actions and aircraft dynamics and audio communications to ATC. The audio was recorded during each flight and analyzed later for specific times the pilot contacted ATC. A table of the outputs provided by the RFS is included in Appendix C. Researcher notes provided information about the pilot's response that may have not been immediately evident through analysis of the RFS data or audio files, such as if the pilot immediately returned to their cleared path after "Clear of Conflict."

Post-Experiment Data Processing

The quantitative data recorded by the integrated simulator is summarized into measures of interest using a Matlab script. The script's algorithms combine the various sources of data (RFS and audio) and scan the entire data set for defined conditions occurring within four time segments: Pre-TA (30 seconds before the TA), after the TA, during the RA, and 60 seconds after Clear of Conflict. The script, in its entireity, can be found in Appendix F.

Because the RFS records a large amount of data for each scenario at varying time intervals (ranging from 0.001 to 0.01 seconds between recorded values), the data is summarized by examining the average, maximum/minimum, difference, and binary (or "Yes/No") values. Measures of interest calculated by the script include defined conditions for autopilot interaction, ATC interaction, vertical rates, turn rates, altitude, heading, and compliance characteristics, as shown in Appendix D.

4.5 Approach to Data Analysis

Two types of statistical tests were used analyze quantitative data: the Mixed Model with Fixed and Random Effects and the Chi-Square Analysis. Qualitative data was analyzed using content analysis. Figure 13 shows a breakdown of the different analyses.

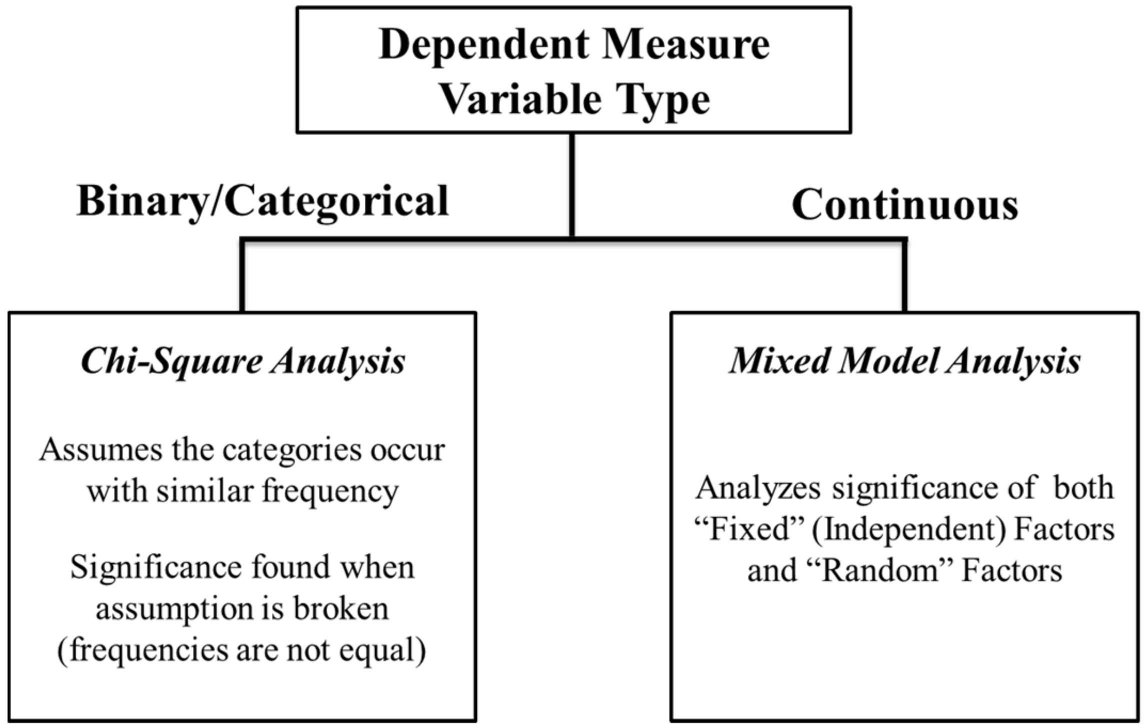


FIGURE 13. Overview of statistical analysis

4.5.1 Binary/Categorical Measures

Chi-Square Analysis was used for categorical data (such as the binary measures). In this type of analysis, the null hypothesis that categorical measures occur with a similar frequency in different conditions such as different traffic events.

4.5.2 Continuous Measures

Mixed Model Analysis

The Mixed Model Analysis was used for analyzing continuous dependent measures. The random factors (or covariates) analyzed for this study were the Run Order and Pilot Effects. For dependent measures where Run Order or Pilot Effects were identified as a significant source of variance, Sidak and Game-Howell pairwise comparisons were made. For pilots effects, those pilots who showed significant differences with greater than 50% of the other pilots were omitted as outliers, as summarized in Appendix E.

Since all events were analyzed simultaneously, significant effects could be due to differences caused by the event structure (ATC information or traffic density) or differences caused by training. Therefore, differences between events were first analyzed by individually comparing the response of each pilot in this study after the training program to each event. Figure 14 has an example of this type of analysis. Second (and separately), each event was isolated and the responses of the trained pilots were compared to the responses of the baseline pilots. Figure 15 has an example of this type of analysis.

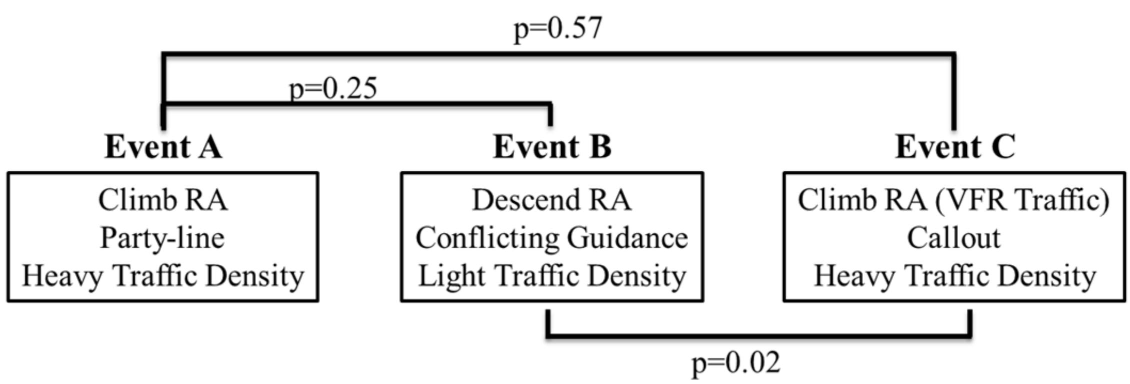


FIGURE 14. Event comparison for statistical analysis

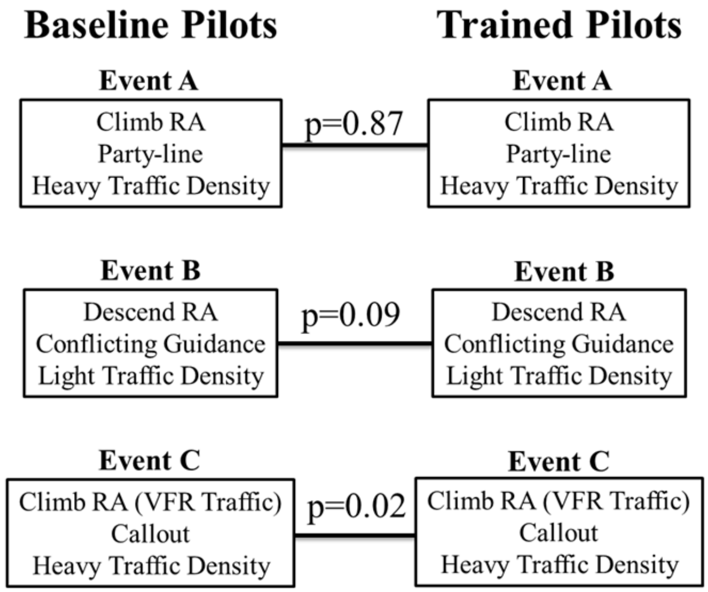


FIGURE 15. Statistical comparison between trained and baseline pilots

Detailed Analysis (Means and Variances)

If a particular event was found in the mixed model to have a significant effect due to training, further analysis examined the equality of the means and homogeneity. The Brown-Forsythe test was used as a robust method of identifying significant differences between population means. Additionally, Levene's Test was used to find significant differences between population variances.

4.5.3 Content Analysis

Pilot responses to the questionnaires were examined for trends and categorized the responses. Further discussion these categories and trends within the comments are in the results.

4.6 Results

The results of the study are summarized below, organized by skill-based, rule-based, and knowledge-based behaviors. Within each behavior level, the response is broken down further into the pilot's response before training, during training, and after training.

4.6.1 Skill-Based Behavior

During-Training Results

As discussed in TCAS literature and in chapter 3, TCAS assumes pilots will initiate a response within five seconds of receipt of a corrective RA and will perform a 0.25 g pull-up (or push-over) to achieve the appropriate vertical rate (FAA 2011a). Thus, an appropriate response to an RA would indicate the pilot disconnecting the autopilot within five seconds of a corrective RA. The time to compliance depends on how much the aircraft must be pitched up or down, but is also on the order of a few seconds. During the EBT events, significant differences were found between the pilots in the training program and the baseline pilots for the skill-based performance measure Time Pilots First Achieved Compliance After RA Initiation. Figure 16 shows that in the event with conflicting guidance the trained pilots complied sooner than the baseline pilots, and their response was more consistent (i.e., had less variance). For the skill-based measure Autopilot Disconnect Time After RA Initiation, no significant difference was observed within any of the training events.

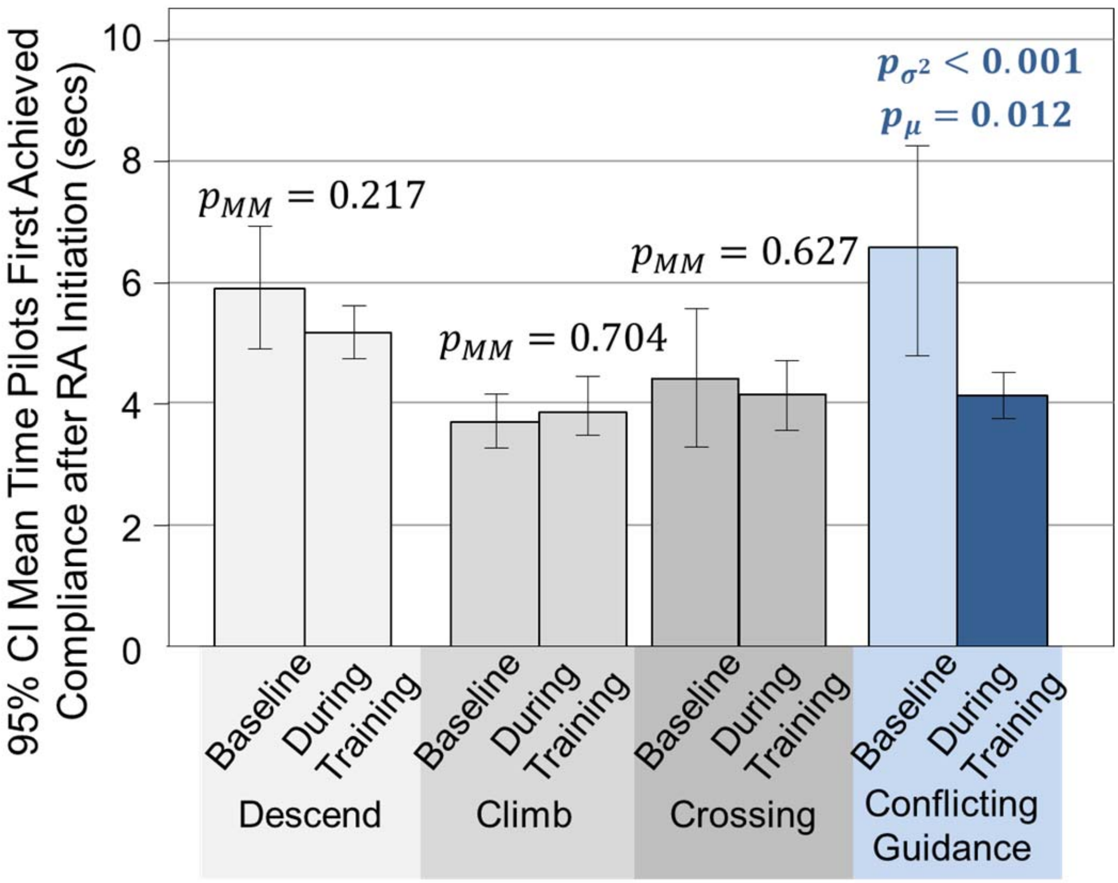


FIGURE 16. Mean and 95% confidence interval of the measure Time Pilots First Achieved Compliance After RA Initiation during EBT events, comparing pilot responses during training to prior baseline study

Post-Training Results

Examining the measures taken in the experiment after the training, both of these skill-based measures were found be significantly different from the baseline pilot. For the mean Time Pilots First Achieved Compliance After RA Initiation, the trained pilots achieved compliance sooner (i.e. lower means) and were more consistent (i.e., lower variance) in their response for the event with conflicting guidance (Figure 17). Also, the trained pilots disconnected their autopilot sooner when compared to the baseline pilots (Figure 18) in two events, and more consistently in the event with conflicting guidance.

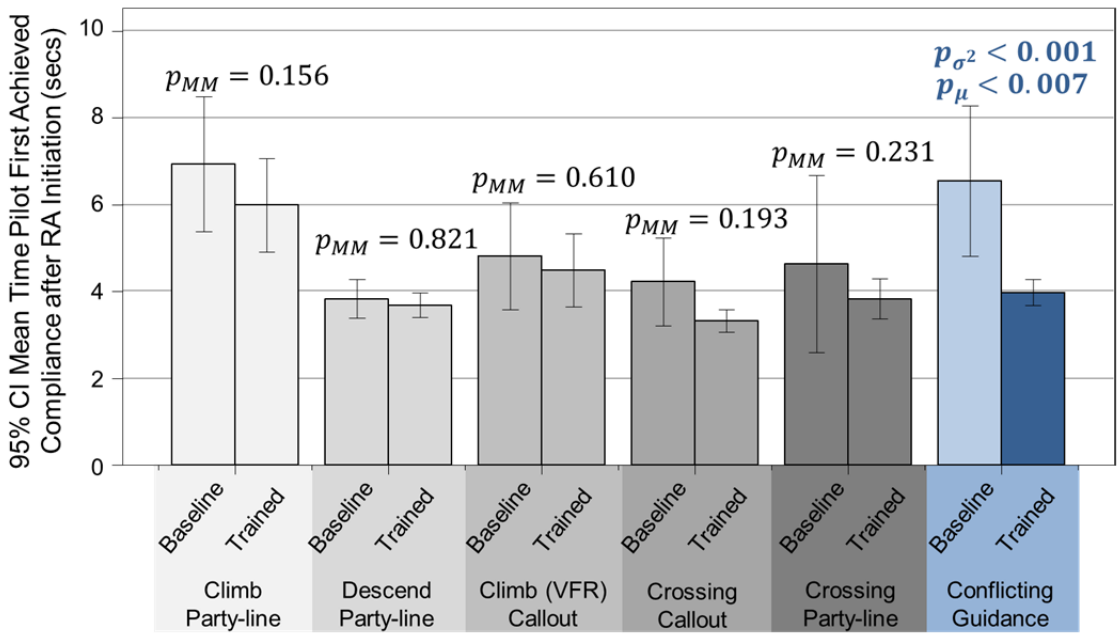


FIGURE 17. Mean and 95% confidence interval of the measure Time Pilots First Achieved Compliance After RA Initiation within each experiment event, comparing trained pilot responses to prior baseline study

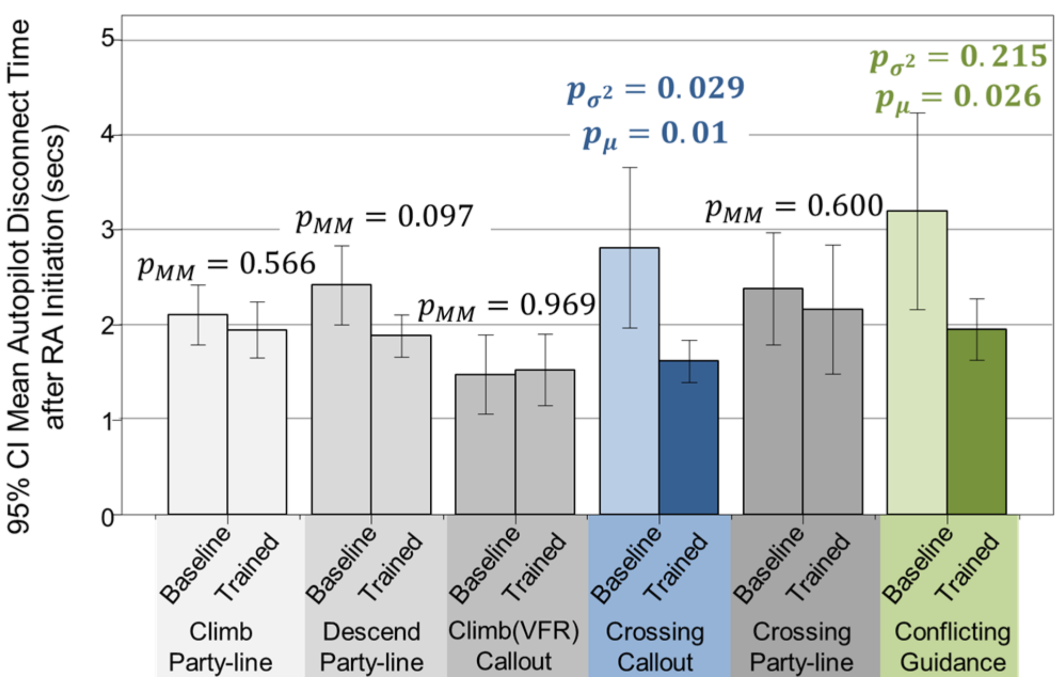


FIGURE 18. Mean and 95% confidence interval of the measure Autopilot Disconnect Time After RA Initiation within each experiment event, comparing trained pilot responses to prior baseline study

4.6.2 Rule-Based Behavior

Pre-Training Results

According to FAA literature, responding to a corrective TCAS RA should typically cause an altitude deviation of no more than 300 to 500 feet with vertical speeds that are not excessive (FAA, 2011a). However, pilot response to the Pre-Training Quiz before completing the training program showed that only two of the eighteen pilot participants (11%) correctly knew that an RA should typically cause less than 500 feet of altitude deviation. Ten of the participants (66%) incorrectly reported they hold the new altitude achieved after responding to an RA. Only thirteen of the pilots (72.2%) were aware that an aggressive maneuver may cause injury to passengers or alter the air traffic flow.

Nearly all pilots (94.4%) knew that compliance to TCAS was achieved by pitching the aircraft out of the red area. One pilot reported he would both pitch the aircraft out of the red and perform a turn away from traffic to comply with TCAS instructions (although TCAS provides no horizontal guidance). For scenarios with conflicting ATC guidance, all eighteen pilots knew to follow TCAS guidance instead of ATC instructions.

The TSD is used to locate nearby traffic, but does not provide any maneuver guidance to the pilot. However, when provided an image of the TSD (Figure 19), with traffic “A,” 500 feet below and climbing, traffic “B” 1100 feet above and descending, and traffic “C” 900 feet below and level, twelve of the eighteen pilots (77%) reported they would have performed some maneuver based solely on this information. Only six (33%) of the pilots responded there was not enough information to maneuver.

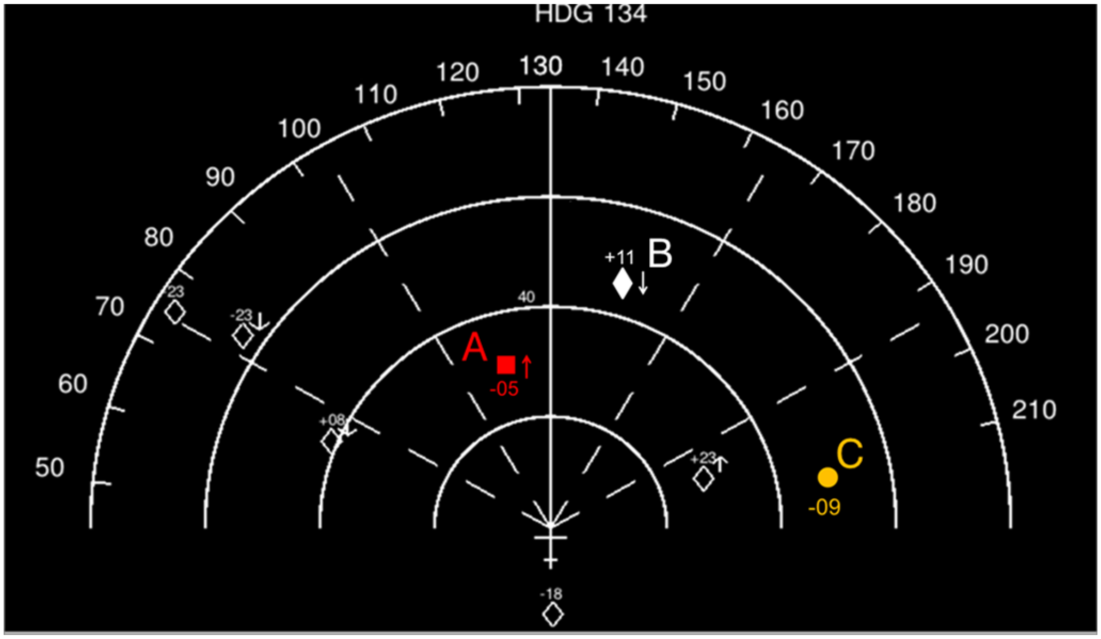


FIGURE 19. TSD shown to the pilots on the Pre-Training TCAS Quiz

In the Pre-Experiment Questionnaire, when asked to describe their airline's procedure for complying with TCAS, six of the eighteen pilots directly called out the need for autopilot disconnect as a part of the established set of procedures. While all eighteen of the pilots noted required compliance with TCAS, five pilots also commented that compliance was not necessary if there was a TCAS malfunction or if the RA would cause an unsafe situation. Fourteen pilots strongly agreed, three pilots agreed, and one pilot was neutral to the statement "I follow the maneuvers displayed by TCAS advisories."

During Training Results

Responding to an RA should typically cause an altitude deviation of less than 500 feet. Additionally, the difference in the TCAS commanded vertical rate and the pilot's vertical rate should be minimized the reduce the impact of the maneuver on air traffic operations. Four of the five EBT events were comparable to the baseline study (the baseline pilots did not receive a Preventive RA). In these four events, the pilots had significantly lower Altitude Deviation Over Duration of RA and the Average Vertical Rate Difference in two of the training events (Climb and Crossing RA's) compared to the baseline study, as shown in Figure 20 and Figure 21. The pilots were also more consistent during the EBT in the Climb RA event in the measure Altitude Deviation Over Duration of RA and in the Conflicting Guidance event in the measure of the Average Vertical Rate Difference. Training was also a significant factor in the measure Percentage Compliance (p=0.005). For this measure, the first training event, Climb RA, had a lower Percentage Compliance compared to the baseline pilots (93.1% compared to 99.6%).

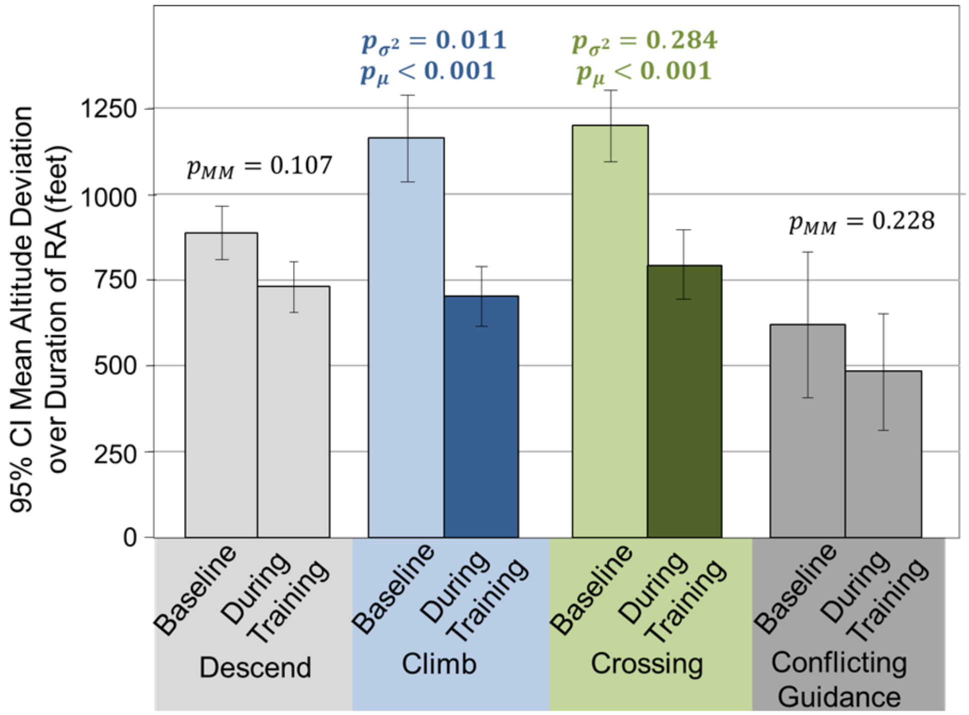


FIGURE 20. Mean and 95% confidence interval of the measure Altitude Deviation Over Duration of RA within each EBT event, comparing pilot responses during training to prior baseline study

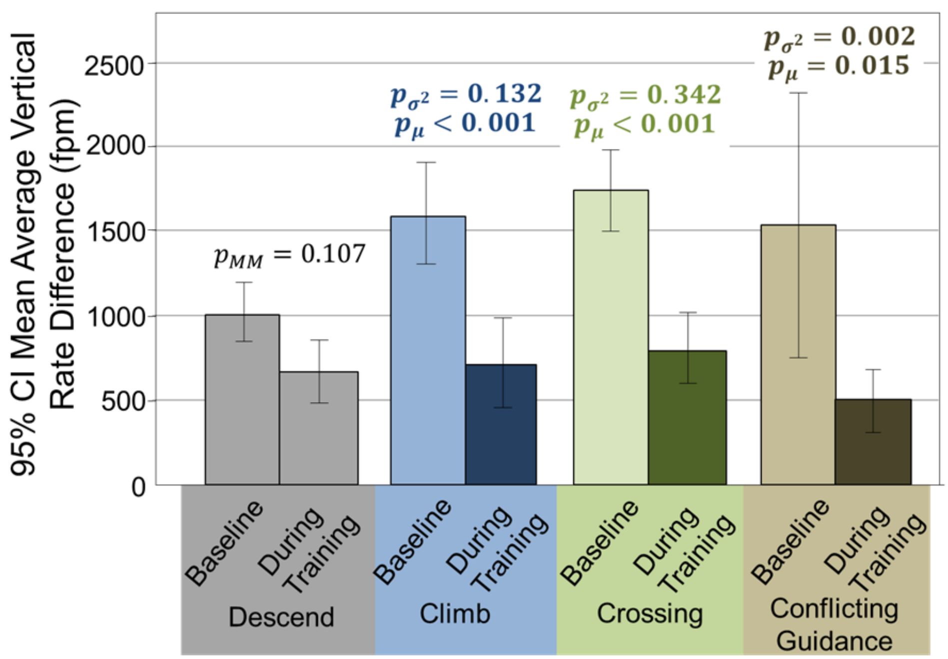


FIGURE 21. Mean and 95% confidence interval of the measure Average Vertical Rate Difference within each training event, comparing pilot responses during training to prior baseline study

Post Training Results

Examining the measures taken in the experiment after the training, aggressive responses were reduced as indicated by nearly all measures for most events, when compared to the baseline study, as shown in Figure 22, Figure 23, Figure 24, and Figure 25. In many cases, the responses were also more consistent within the trained pilots compared to the baseline, i.e. the variance was also significantly lower.

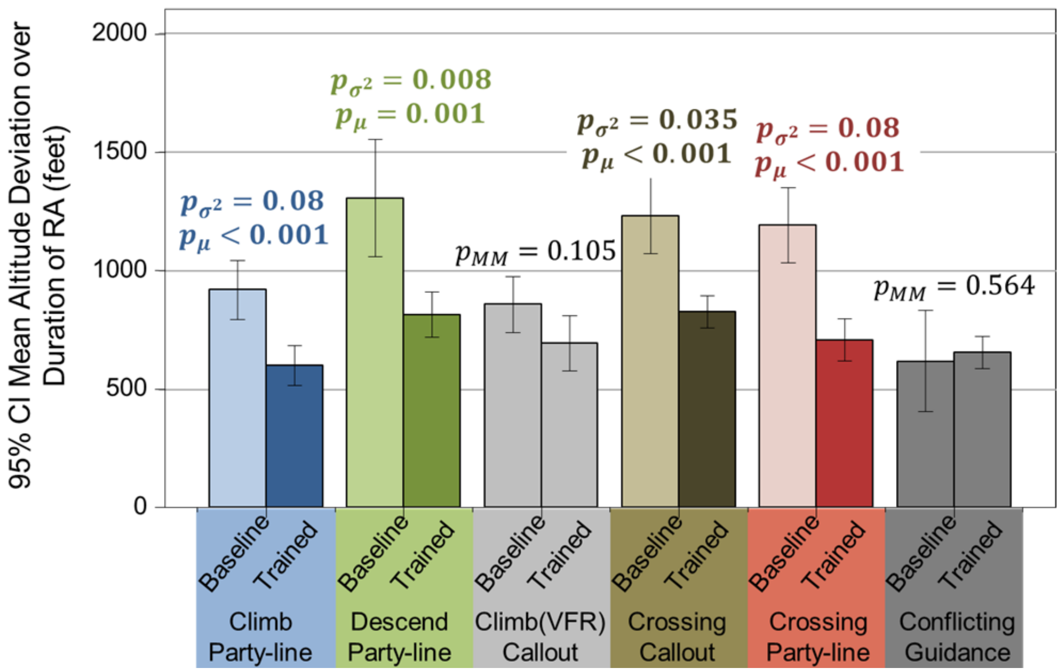


FIGURE 22. Mean and 95% confidence interval of the measure Altitude Deviation Over Duration of RA within each experiment event, comparing trained pilot responses to prior baseline study

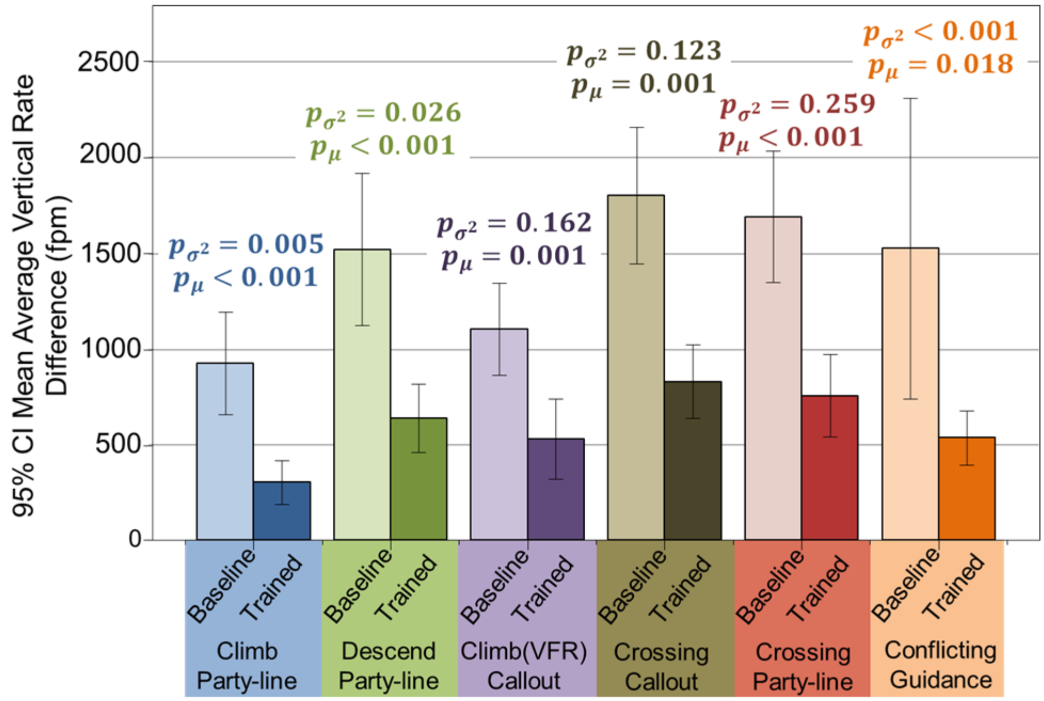


FIGURE 23. Mean and 95% confidence interval of the measure Average Vertical Rate Difference within each experiment event, comparing trained pilot responses to prior baseline study

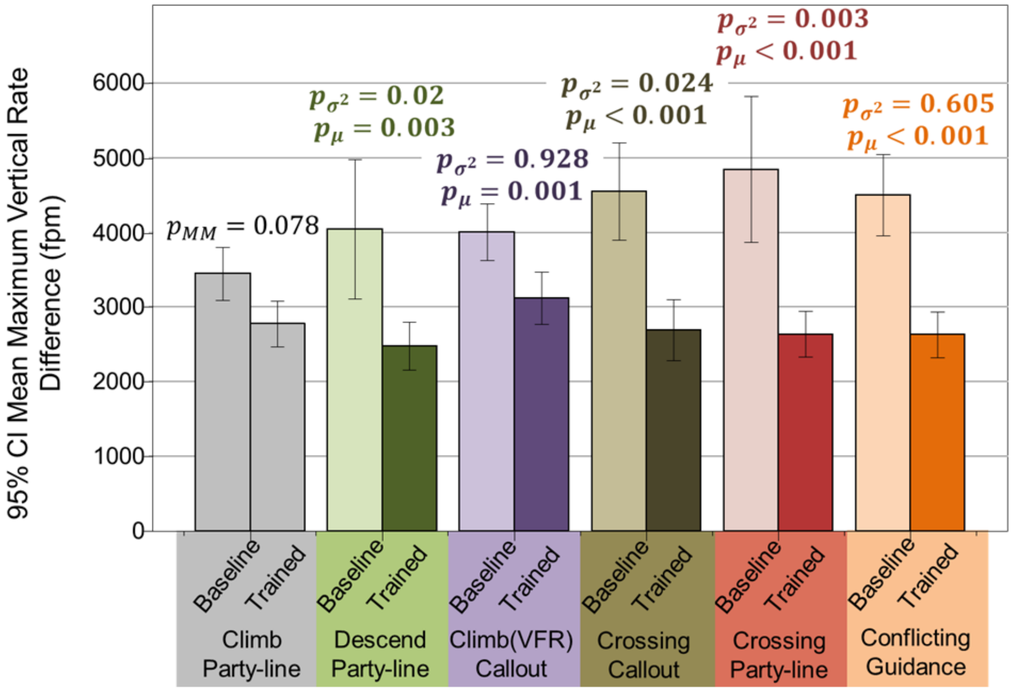


FIGURE 24. Mean and 95% confidence interval of the measure Maximum Vertical Rate Difference within each experiment event, comparing trained pilot responses to prior baseline study

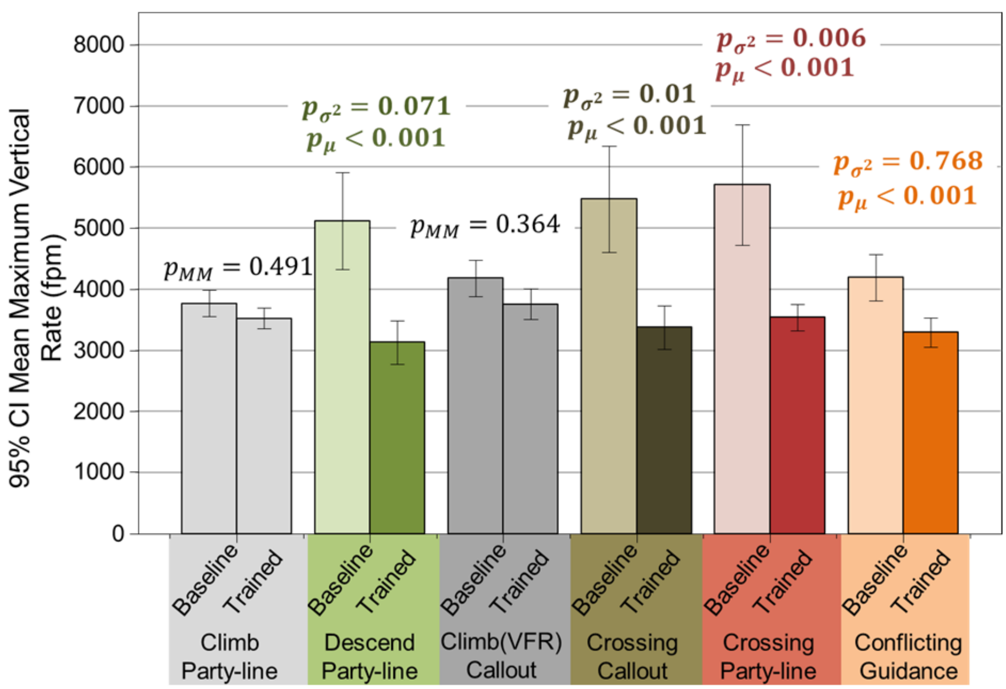


FIGURE 25. Mean and 95% confidence interval of the measure Maximum Vertical Rate within each experiment event, comparing trained pilot responses to prior baseline study

On the Post-Experiment Questionnaire, five pilots directly commented about the need for monitoring aggressive responses of the response. In response to the statement “I am less likely to perform excessive maneuvers in response to RA’s after completing today’s training program,” ten pilots agreed, four pilots were neutral, and four pilots disagreed.

In the baseline study the event with conflicting guidance had been notable for significantly larger unconsistency (i.e., variance); after the training program, the trained pilots were more consistent in this event in their compliance as captured by the measure Percentage Compliance (Figure 26). Furthermore, for the measure Pilot Response After Clear of Conflict, when compared to the baseline pilots, the trained pilots more often returned to their previous clearance and did not contact ATC for further instructions or a new clearance (Figure 27). However, when examining Pilot Interaction With ATC after the TA and after the RA, it was observed that trained pilots contacted ATC less often than the baseline pilots (Figure 28 and Figure 29).

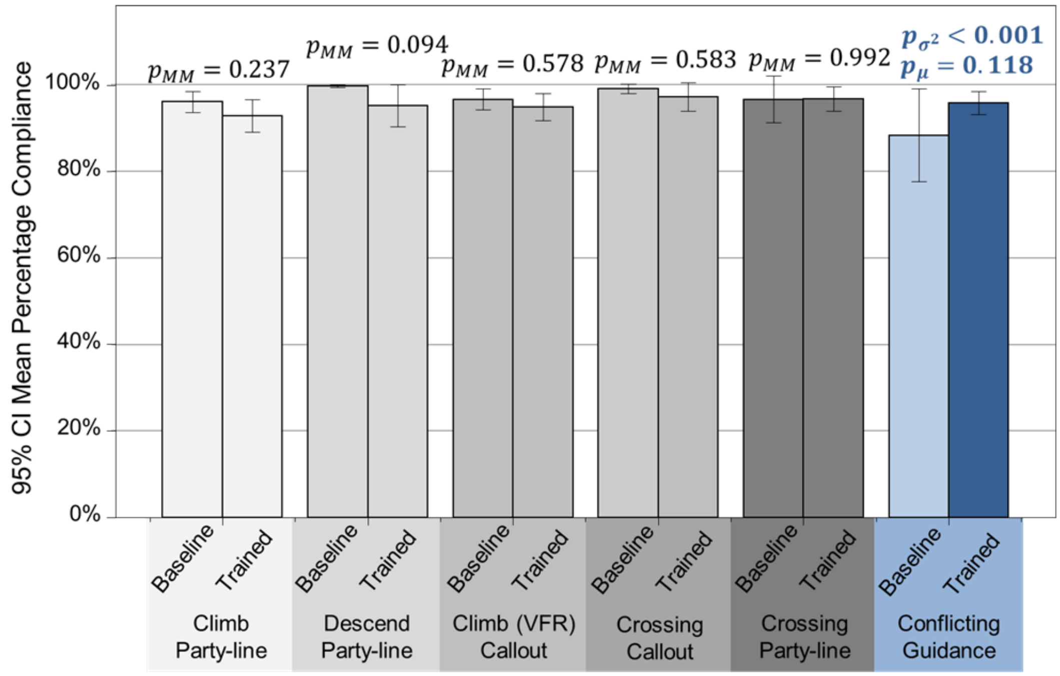


FIGURE 26. Mean and 95% confidence interval of the measure Percentage Compliance within each experiment event, comparing trained pilot responses to prior baseline study

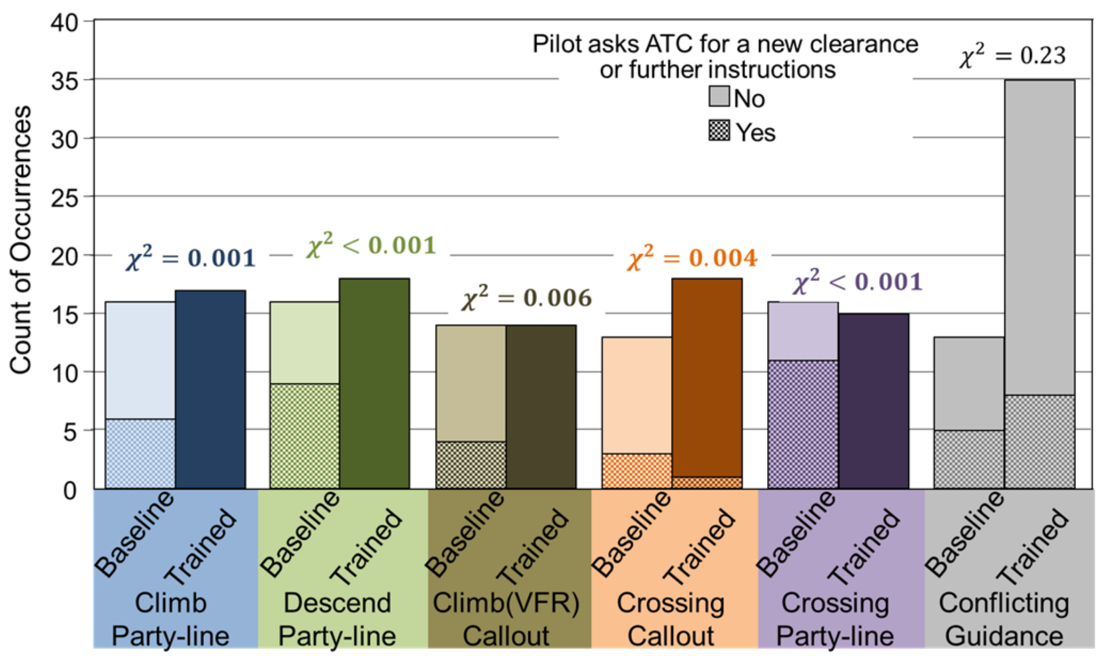


FIGURE 27. Pilot Response After Clear of Conflict: Comparison of the frequency at which the baseline and trained pilots' contacted ATC for further instructions after the RA

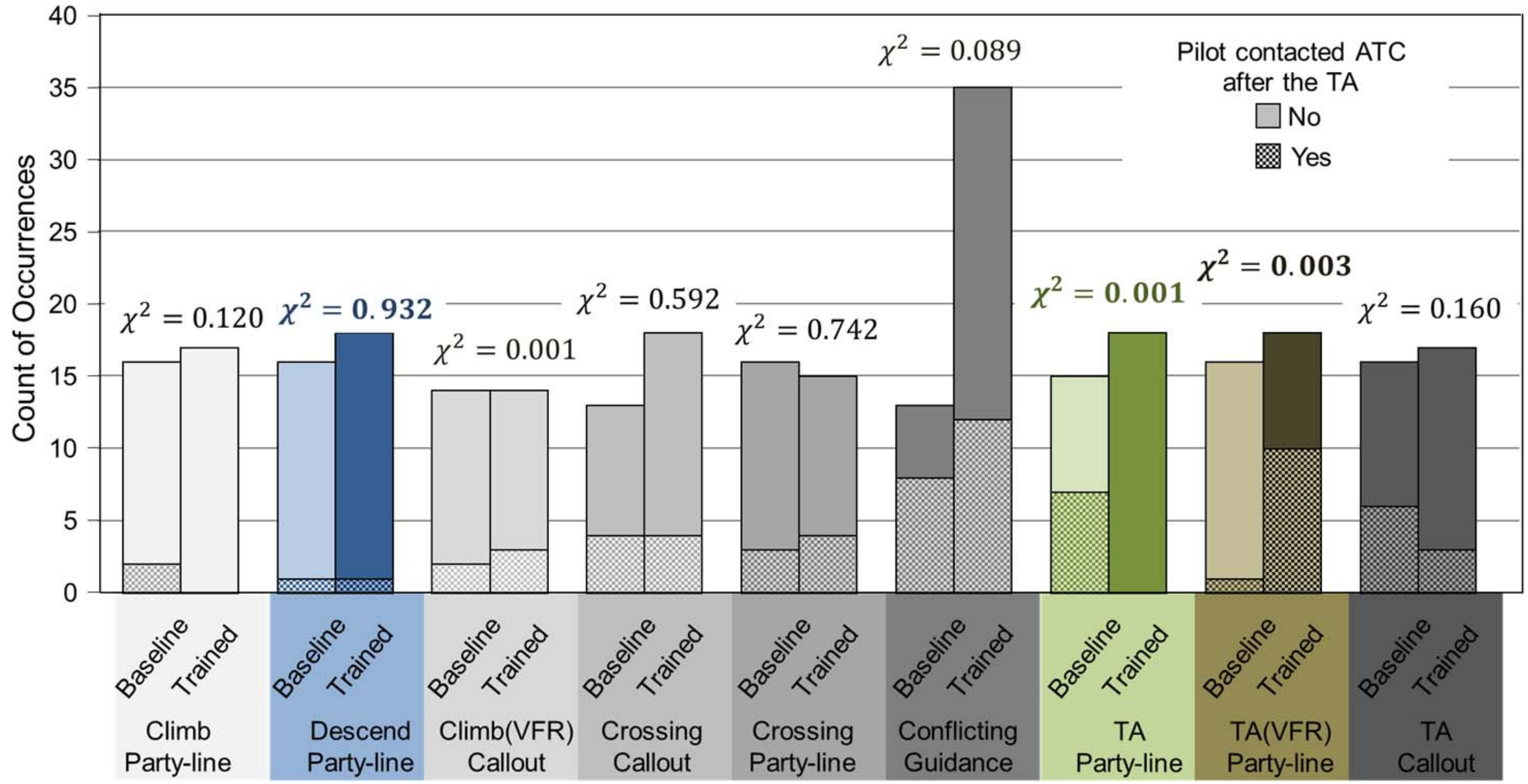


FIGURE 28. Pilot Interaction With ATC after the TA: Comparison of the frequency at which the baseline and trained pilots' contacted ATC after the TA

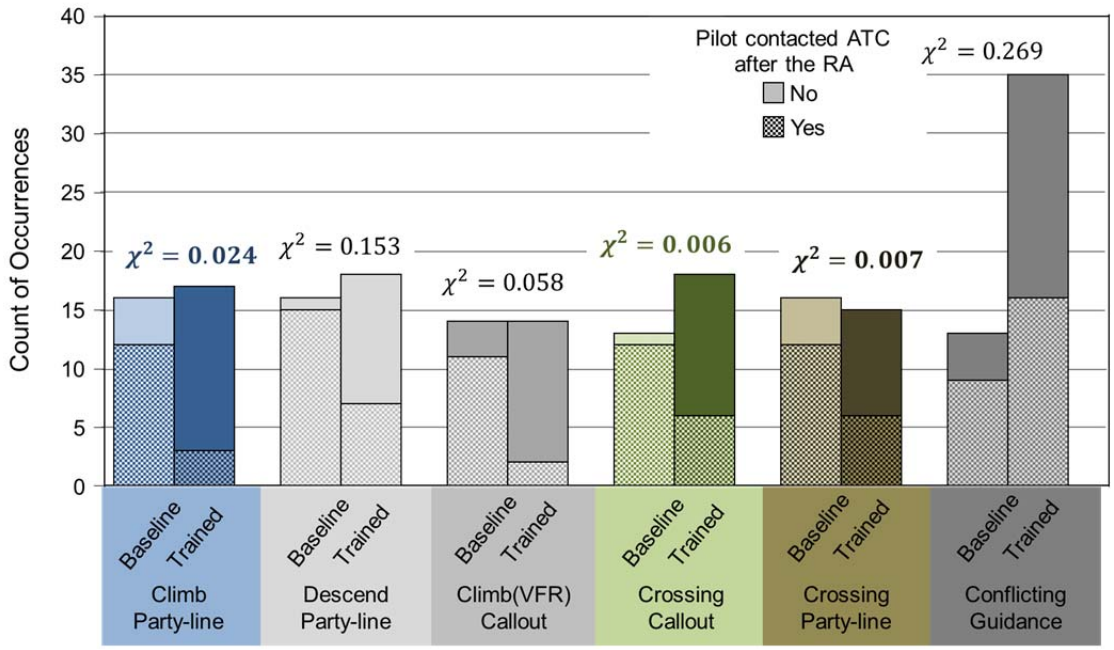


FIGURE 29. Pilot Interaction With ATC after the RA: Comparison of the frequency at which the baseline and trained pilots' contacted ATC after the RA

4.6.3 Knowledge-Based Behavior

Pre-Training Results

In responding to the Pre-Experiment Questionnaire, seventeen of the eighteen pilots strongly agreed and the remaining one pilot agreed with the statement "I understand TCAS maneuvers when they are issued." However, results of the Pre-Training Quiz found that, while some pilots exhibited adequate TCAS knowledge, others knew very little about TCAS advisory logic and constructs of the traffic environment. When asked about assumptions made by TCAS advisory logic, nine of the eighteen pilots (50%) correctly identified the assumptions. Two questions asked pilots to interpret symbols on the TSD, and 9 of the pilots (50%) correctly answered both questions related to TSD symbology.

Post Training Results

In the Post-Experiment Questionnaire, six pilots strongly agreed, eleven pilots agreed, and one pilot was neutral with the statement “My understanding of TCAS has increased.” In the open response questions, nine of the participants directly commented on an increase in understanding TCAS logic. Five of the eighteen pilots claimed to have learned about the different types of RA’s (notably the Crossing RA), with one pilot responding “Types of RA's was not previously taught. We were taught simply to comply.” Two pilots strongly agreed, seven pilots agreed, and seven pilots were neutral, and two pilots strongly disagreed with the statement “I am more likely to trust TCAS after completing today's training”. One pilot who commented they strongly disagreed with the statement later remarked “My trust in TCAS was already at maximum so I wouldn't be ‘more’ likely to trust it.” In the open response questions, three pilots directly called out an increase in trust of TCAS.

Results related to Pilot Interaction With ATC showed the trained pilots were more likely to contact ATC before the TA in the scenario with conflicting ATC guidance (Figure 30). However, the pilots were less likely to contact ATC before the TA in event with a Climb RA caused by VFR traffic as well as in the TA-only event with party-line information.

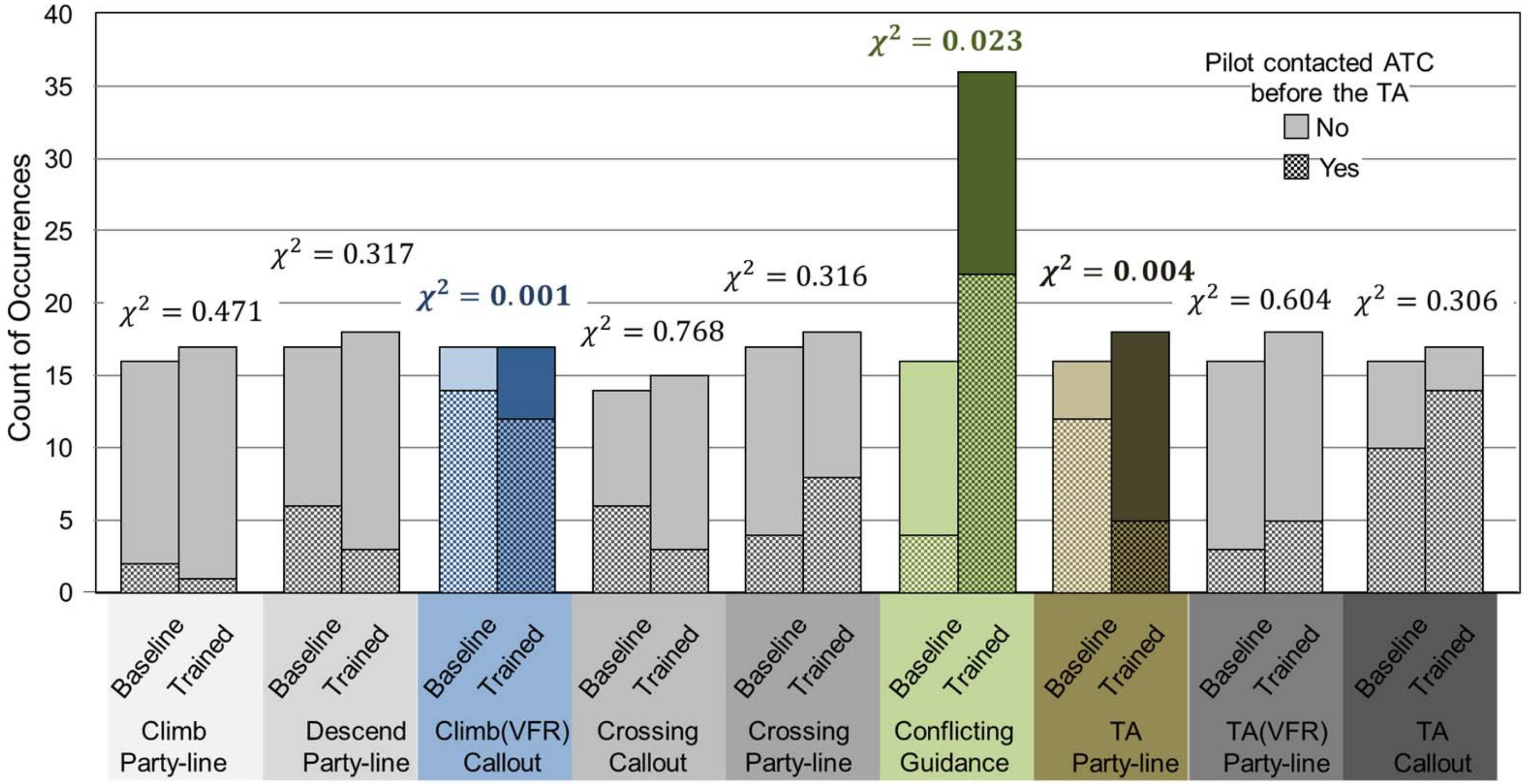


FIGURE 30. Pilot Interaction With ATC: Comparison of the frequency at which the baseline and trained pilots' contacted ATC before the TA

4.7 Summary of Results

Effectiveness of the training program was measured by evaluating its impact, i.e. whether the training objectives were reflected in pilot performance at the skill, rule, and knowledge-based behavior levels. Examining skill-based behaviors, the TCAS training is intended to decrease the Time Pilots First Achieved Compliance and the Autopilot Disconnect Time After RA Initiation. These measures were of particular concern in the baseline study for specific events, notably the event with conflicting ATC/TCAS guidance and the event with a Crossing RA, which had both a large means and large variances. For the measure Time Pilots First Achieve Compliance, decreases in the variance due to the training were noted starting as early as the EBT training events. Furthermore, when comparing the baseline to the trained pilots, decreases in both the mean and variance were noted for the conflicting guidance scenario. For the measure Autopilot Disconnect Time, decreases in the mean due to the training were noted for both the event with a Crossing RA and the event with conflicting guidance, and a decrease in the variance was also noted for the event with a Crossing RA.

Examining rule-based behaviors, before completing the training program, a majority of pilots would have incorrectly performed a maneuver based solely on information provided by the TSD. Only two of the eighteen pilots (11%) knew the correct amount of altitude deviation an RA is expected to cause and twelve of the eighteen participants (66%) would not immediately return to their previously cleared path after responding to an RA. Seventeen of the eighteen pilots (94.4%) knew compliance was achieved by pitching out of red area. These reflect misconceptions about the rules for proper interaction with TCAS.

Thus, the training program was intended to improve measures of proper rule-based behavior in response to TCAS advisories. Examining compliance, there was a decrease in variance for the event with conflicting guidance. Examining measures of aggressive responses, even as compliance stayed the same or improved, the aggressiveness decreased (improved). This improvement was noted as early as the EBT training events in some measures. Furthermore, in examining all four aggressive measures, decreases in both the mean and variance due to training were noted for most traffic events after completion of the training program.

Although not required for compliance, training objectives sought to support knowledge-based behavior related to the pilot's ability to recognize and predict events. As an example of gaps in knowledge, before completing the training program, nine of the eighteen pilots (50%) could not correctly identify all of the traffic symbols on the TSD. After training, most pilots believed their understanding of TCAS increased with nine of the eighteen pilots (50%) of the participants directly commenting on their increased understanding of TCAS logic. Furthermore, nine of the eighteen pilots (50%) of the pilots said they were more likely to trust TCAS after training program.

CHAPTER 5

CONCLUSIONS

5.1 Summary

The objectives of this thesis were to:

- train pilots to understand TCAS use for collision avoidance in the actual traffic and operational environment, and
- provide pilots with a well-rounded knowledge of different traffic situations that may result in TCAS advisories.

These objectives were accomplished by developing a TCAS training program integrating Demonstration Based and Event Based Training techniques. This integration enhances skill-based, rule-based, and knowledge-based behaviors. The program's training objectives follow those required by FAA regulatory material and also add a training objective intended to reduce aggressive responses to TCAS RA's. The DBT segment of the training program provides an interactive interface, mid-training quizzes, and a concluding demonstration of the evolution of a traffic situation. The overall design of the DBT segment was intended to engage the pilot in the learning process and fully portray the important signs and symbols that correspond to the intended rule-based and knowledge based behaviors. The EBT segment of the training program requires an overt action for successful task completion. Skill-based and rule-based behaviors are enhanced through performance of associated rules, and knowledge-based behaviors are reinforced through a broad experience with TCAS events made possible by the inclusion of a realistic environmental context.

The efficacy of the training program was assessed in an integrated ATC-cockpit simulator study, in which eighteen pilots first completed the modified training program and then participated in a flight simulator study. Performance of the trained pilots was compared to the performance of sixteen baseline pilots who had not receive the modified training program in a previous experiment with the same traffic events. The training objectives defined the measures of performance.

Overall, the training program had a significant impact on the pilots' behavior and response to TCAS advisories. In measures of pilot response in terms of compliance and aggressiveness, significant improvements were found in both the consistency and/or the mean. Pilot response was of particular concern in a baseline TCAS study for specific events, notably the event with conflicting ATC/TCAS guidance and the event with a Crossing RA, for which measures exhibited both a large mean and large variance. After completing the training program significant improvements were found in both the variance (i.e. response consistency) and the mean of the trained pilots' responses. Finally, on the post-experiment questionnaires, pilots commented on their increase in understanding of TCAS as well as an increase in their trust in the advisory system.

5.2 Contributions

These results contribute to the literature available on methods for designing an effective training program for safety critical systems within common constraints on the program's duration. Currently, pilots are expected to complete substantial training on aircraft systems, in addition to several hundred hours as Pilot in Command of an aircraft, before being allowed to fly with passengers aboard, even for minimal qualifications as the First Officer on regional jets. The significant amount of training already required for pilots to be competent on their aircraft systems suggests that pilots may not need *more* training but instead need *better* training. Currently, pilots complete ground-based training and flight training segments separately, and often the goals of the two training segments do not relate. By building the segments together using DBT and EBT, ground-based training and flight training can be better integrated and help build topics from the classroom to a realistic environment. Additionally, the results of this thesis show that, by creating EBT events in an integrated ATC-flight simulater, pilots can be trained to a desired level within the same amount (if not less) time.

Additionally, this research adds to the discussion regarding current TCAS training objectives outlined by the FAA in Aircraft Circular (AC) 120-55C. As evident through reported injuries caused by zealous pilots, response to TCAS RA's should be sufficient but not excessive. Appendix 6 in AC 120-55C includes language related to understanding the various RA types and thresholds. However, language is also needed to stress the potential negative impacts of aggressive responses. Specifically, section 2 "TCAS Ground Training" presents general concepts of TCAS operation for pilots to understand and include basic concepts of TCAS logic, closest point of approach (CPA), tau, altitude separation thresholds for the issuance of RAs, as well as the relationship between displayed traffic information and issuance of TAs and RAs. "TCAS Ground Training" could also include language to address the potential consequences of excessive responses to TCAS RAs. Additionally, language should be added to directly call out the maximum altitude deviation generally caused by an RA (300 to 500 feet).

In section 3 of AC 120-55C, “Classroom Training,” the criteria of advisory thresholds (3.b.2) should also include the criteria for the issuance of a weakening RA; i.e. weakening RAs are given when a safe vertical separation is achieved before a safe range separation and are intended to minimize the aircraft’s altitude deviation. For flight training, language is already included focusing on pilot responses to weakening RAs. However, in the implementation of TCAS flight training, instructors should stress the reduction of the aircraft’s vertical speed with a weakening RA.

Furthermore, as revealed in the pre-training results, pilots are not reflecting the minimal knowledge required by the FAA training standards. For example, many pilots expressed that they had not been previously taught the various RA types. This result may highlight areas for improvement in the clarity and specificity of the training standards.

5.3 Future Research

While this research examines the implementation of a training program in a highly controlled environment, future research should assess its implementation. A pragmatic approach to the implementation of this type of training program would ascertain the overall cost of the program as well as the required improvements to current facilities (e.g. an ATC-cockpit integrated simulator). Furthermore, while the instructor in this program was able to provide performance feedback in real-time a computer could serve the role as instructor for quick performance assessment. Such a digital instructor would be able to provide feedback using quantitative measures similar to those analyzed in this study.

This thesis examined training in the context of pilot response to TCAS advisories; however future work may consider the implications of training design versus system design. While a perfect world would offer an infinite amount of time for training on all subsystems, pilots have neither the time nor ability to remember and appropriately apply knowledge for all aircraft subsystems in every context. Instead, engineers and system designers must consider the implications of system complexity relative to the limited duration available for pilot training, and relative to the practical limits on retention, both in terms of duration and ability to remember fine distinction in concepts and rules for effective responses.

Finally, results of this research extend into broader questions related to pilot interaction with TCAS and ATC. Pilot compliance to an RA is mandated by the FAA unless the pilot believes that the maneuver would endanger safe flight operations. A pilot is authorized to disagree with TCAS and chose to not follow its advisory when it malfunctions or its advisory is considered unsafe. Therefore, the question that arises is "How do we train pilots to recognize when TCAS is wrong?" Another question that arises from this research is related to pilot interaction with ATC during a traffic event. Currently, pilots are only required to contact the controller if an ATC clearance or instruction is volated. Thus, "Are there instances when the pilot should contact ATC even if the clearance is not violated?" The reflection here is if it would be beneficial to inform ATC of the advisory. These questions, although loosely addressed in the current training standards, currently have no codified answer. Therefore, it may be valuable for the future research to address those questions in a more definitive manner.

APPENDIX A: TCAS TRAINING PROGRAM SLIDES

The Traffic alert and Collision Avoidance System

Contents

Introduction to TCAS

Traffic Situation Display

TCAS Advising Logic

Traffic Advisories

Resolution Advisories

Example Timeline of a TCAS Event

TCAS Simulator Training Scenarios

Contents

Introduction to TCAS

Contents

What is TCAS?

"Last line of defense" alerting system intended to prevent mid-air collisions

Traffic Advisory (TA)
An initial advisory warning to direct your attention

Resolution Advisory (RA)
If the situation persists, displays a vertical collision avoidance maneuver

Contents

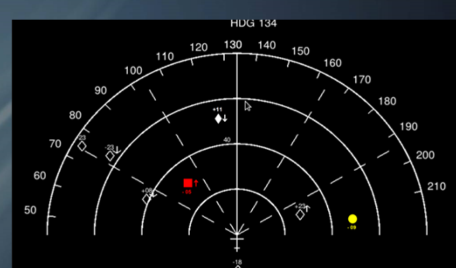
Traffic Situation Display

Contents

Horizontal plan view of nearby aircraft

Numeric Text

- o Relative (or absolute) altitude
- o Assume relative altitude for this training program



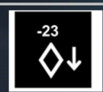
Arrow

- o Relative vertical speed
- o Shown if trend is greater than 500 fpm

Contents

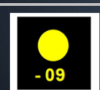
Symbology

Own-aircraft: Airplane-like symbol in white or cyan



Other Traffic: Unfilled diamond in white or cyan

Proximate Traffic: Filled diamond in white or cyan.
Traffic located within 6 nm horizontally and 1200 feet vertically.



Traffic Advisory (TA): Filled yellow/amber circle.
Accompanied by an aural "Traffic, Traffic"

Resolution Advisory (RA): Filled red square.
Accompanied by aural and visual vertical maneuver instructions

Contents

Mid-Training Quiz

Take a moment to interpret the symbols below, when you know the status of aircraft A,B,C, and D click "Answers"

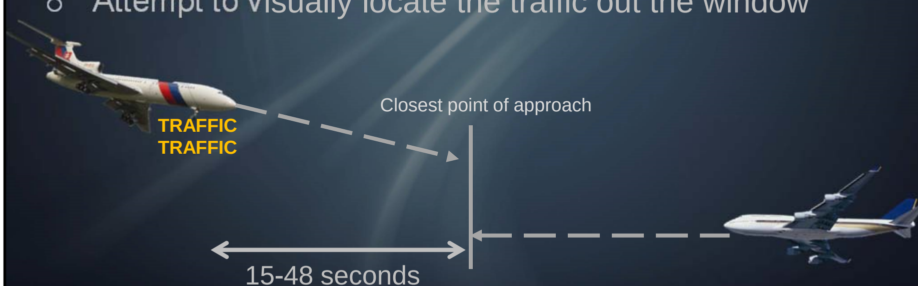
A: Other traffic, 2500 feet above, descending
B: TA, 1000 feet above, descending
C: RA, 300 feet below, climbing
D: Proximate traffic, 1500 feet above, level (not climbing or descending greater than 500 fpm)

Answers

The Traffic Advisory

Traffic Advisory (TA)

- Approximately 15-48 secs from closest point of approach or within a minimum separation
  - Alert thresholds vary with altitude
- Do not maneuver outside of ATC clearance
- Attempt to visually locate the traffic out the window



Contents

Aural:
"TRAFFIC, TRAFFIC"

Visual:
Amber/Yellow circle on Traffic Situation Display

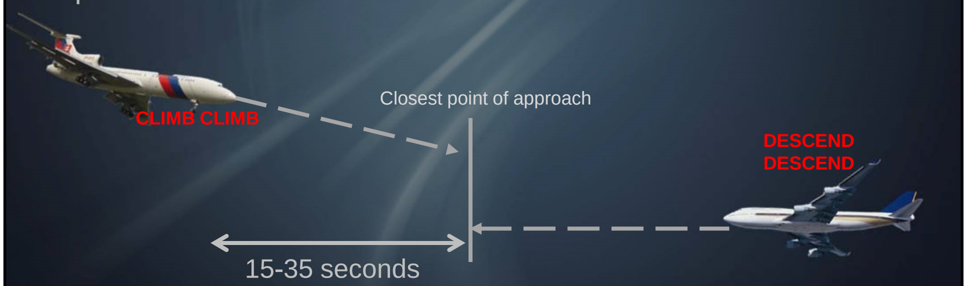
Contents

Resolution Advisory

Contents

Resolution Advisory (RA)

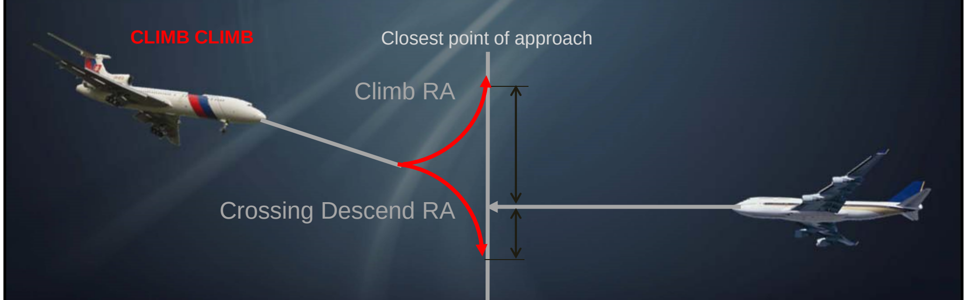
- Calculated using time until closest point of approach (15-35 secs) and a minimum separation
  - Alert thresholds vary with altitude
- Maneuver provides at least 300 feet of vertical separation if performed within 5 seconds



Contents

Sense Selection

- Upward and downward sense RA's are compared
- RA sense providing greatest vertical separation selected
- If other aircraft is also TCAS equipped, it may be given the opposite sense

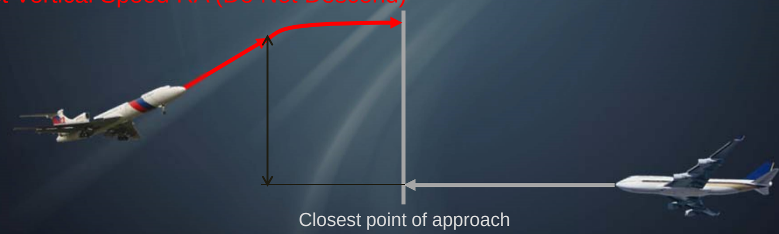


Contents

Weakening RA's

- Initial RA is weakened if safe vertical separation is achieved before range separation is achieved
- Minimizes displacement from original altitude
- Response expected in 2 ½ seconds

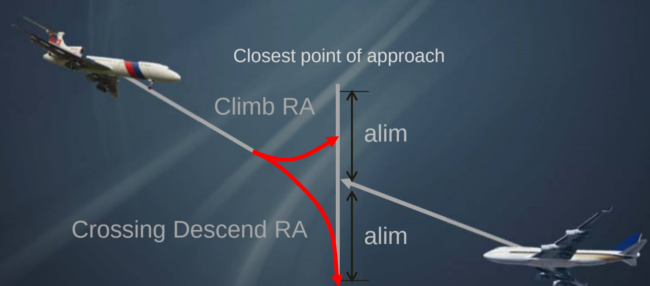
Adjust Vertical Speed RA (Do Not Descend)



Contents

Crossing RA's

- Non-crossing RA's are preferred if they can provide minimum vertical separation (alim)
- Typically occur when aircraft are climbing or descending with high vertical rates

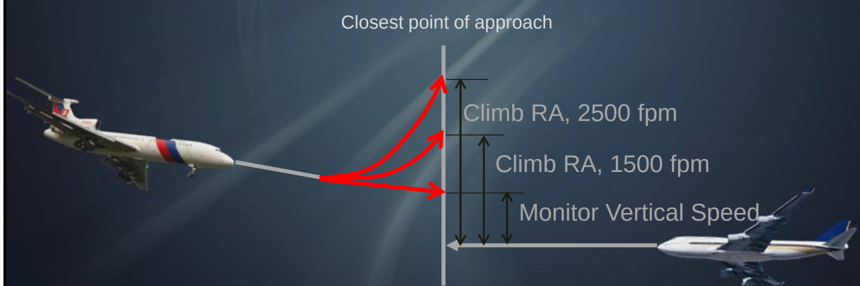


Only the crossing descend RA provides alim separation in this case

Contents

Strength Selection

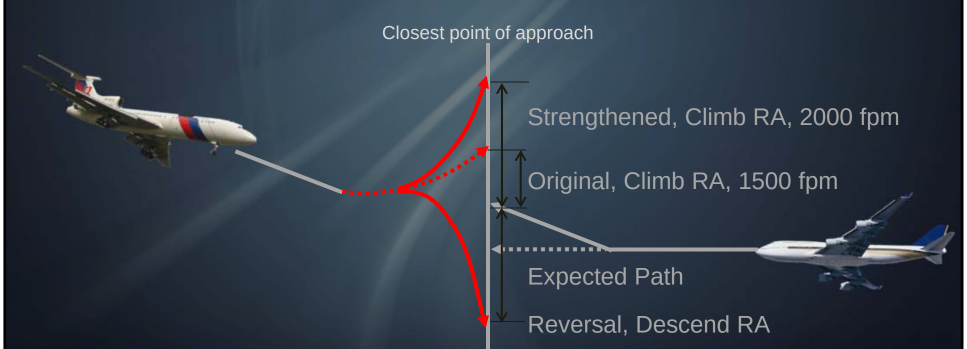
- Selects RA strength least disruptive to existing flight path that provides minimum vertical separation



Contents

Strengthening and Reversal RA's

- If intruder vertically maneuvers opposite to expected, RA may strengthen or reverse direction
- TCAS assumes a 2.5 second response time



Contents

RA Types

Click the RA type to hear its audio and read a short description

Corrective
- Climb
- Descend
- Maintain Vertical Speed

Crossing
- Crossing Climb
- Crossing Descend

Reversal
- Climb NOW
- Descend NOW

Strengthening
- Increase Climb
- Increase Descent

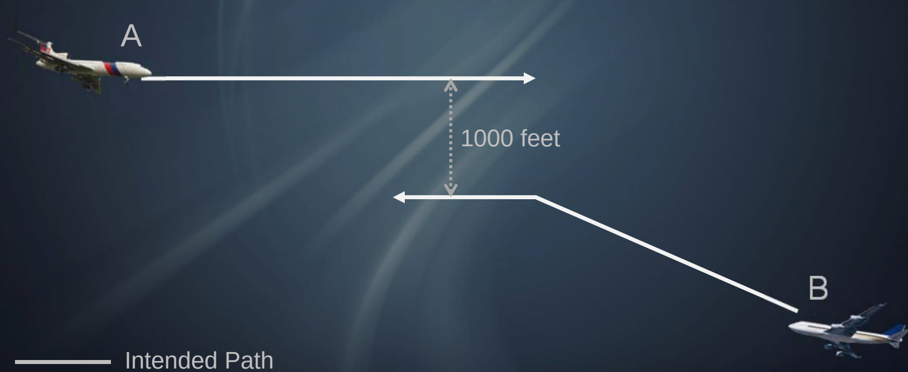
Weakening RA
- Adjust Vertical Speed

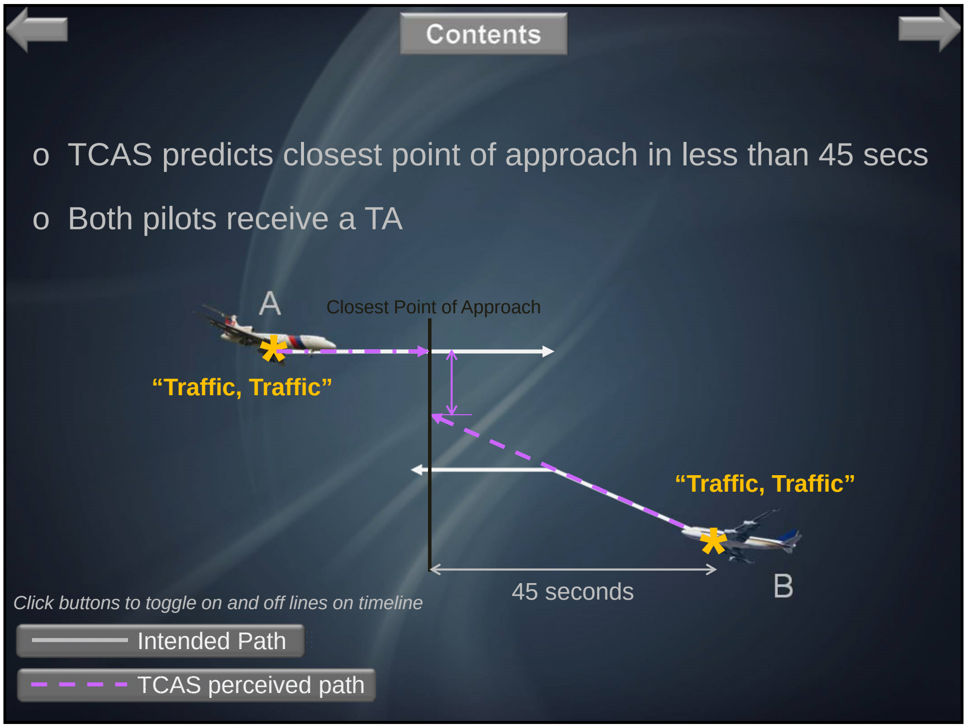
Preventive
- Monitor Vertical Speed

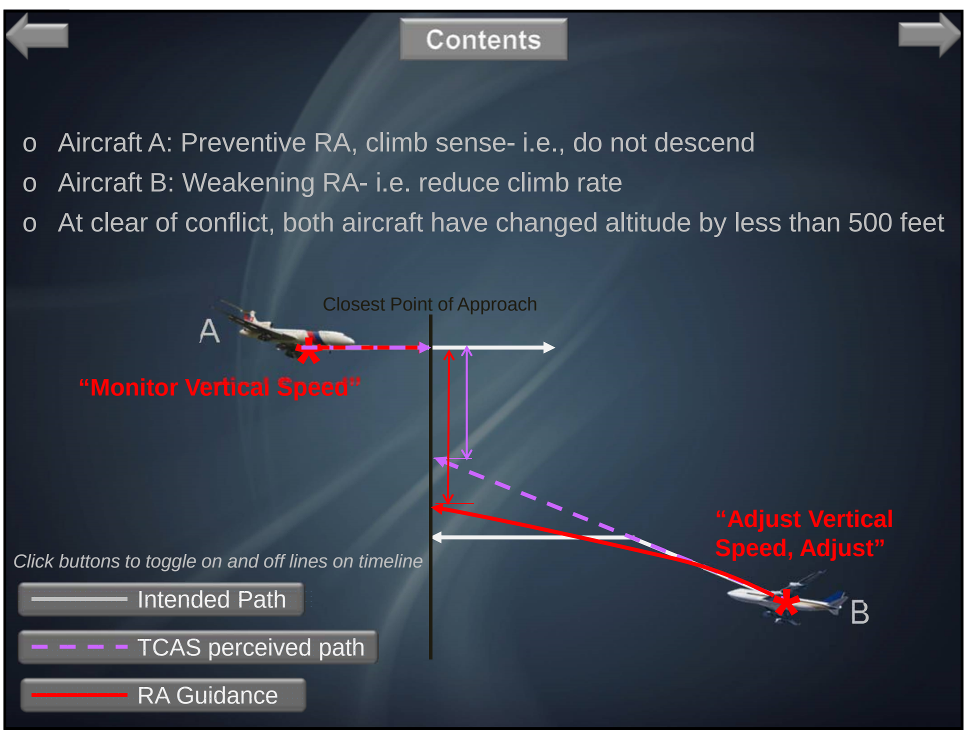
Contents

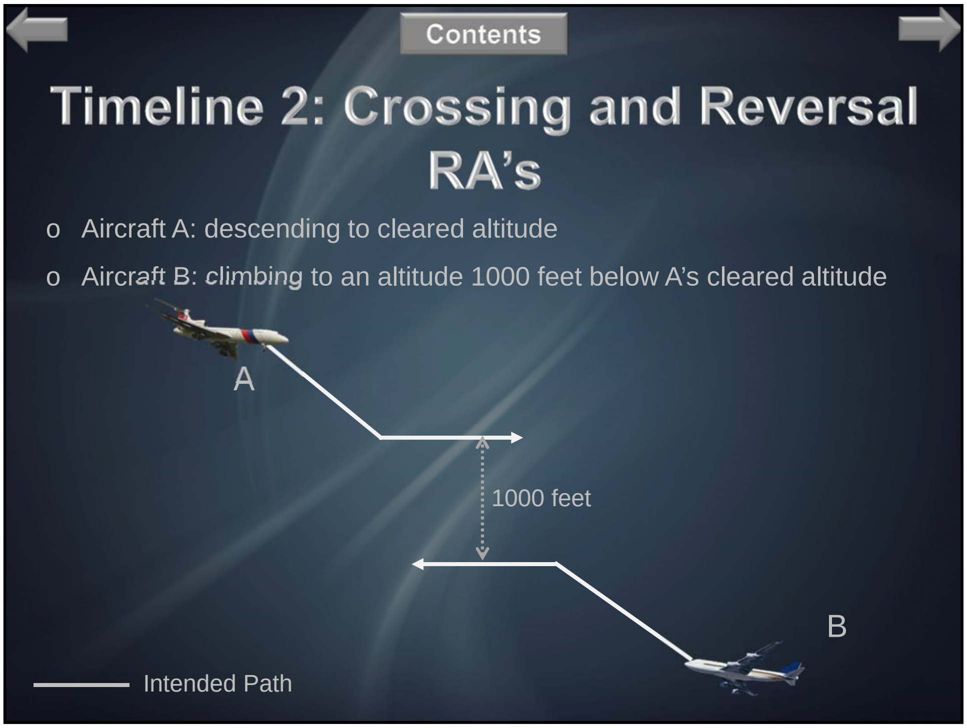
Timeline 1: General RA

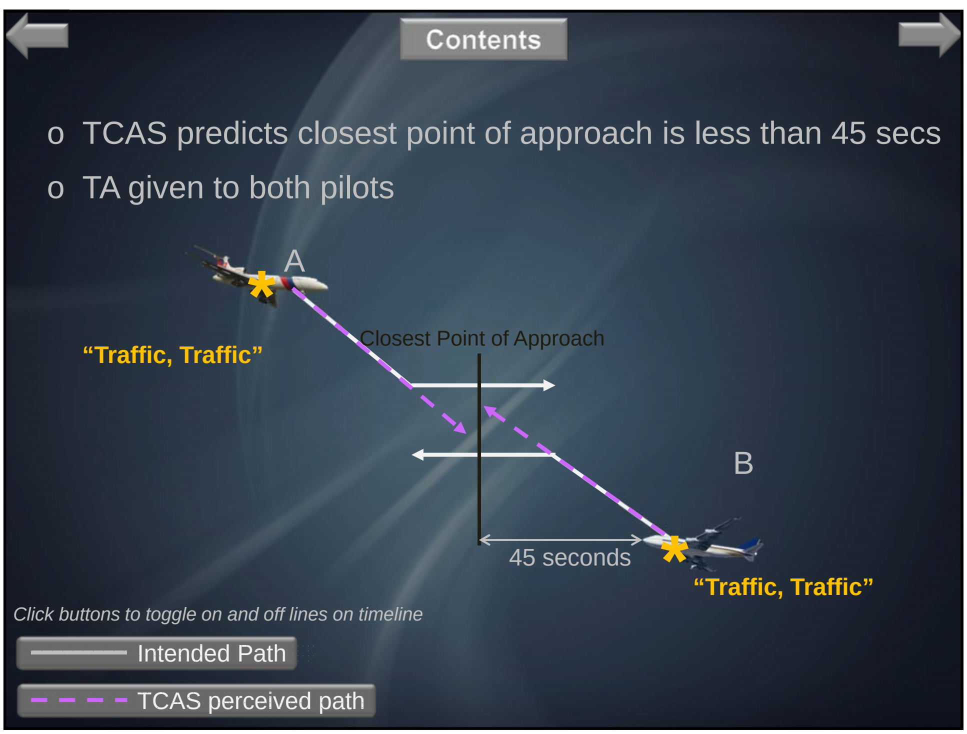
- Aircraft A: level
- Aircraft B: climbing to a cleared altitude 1000 feet below aircraft A

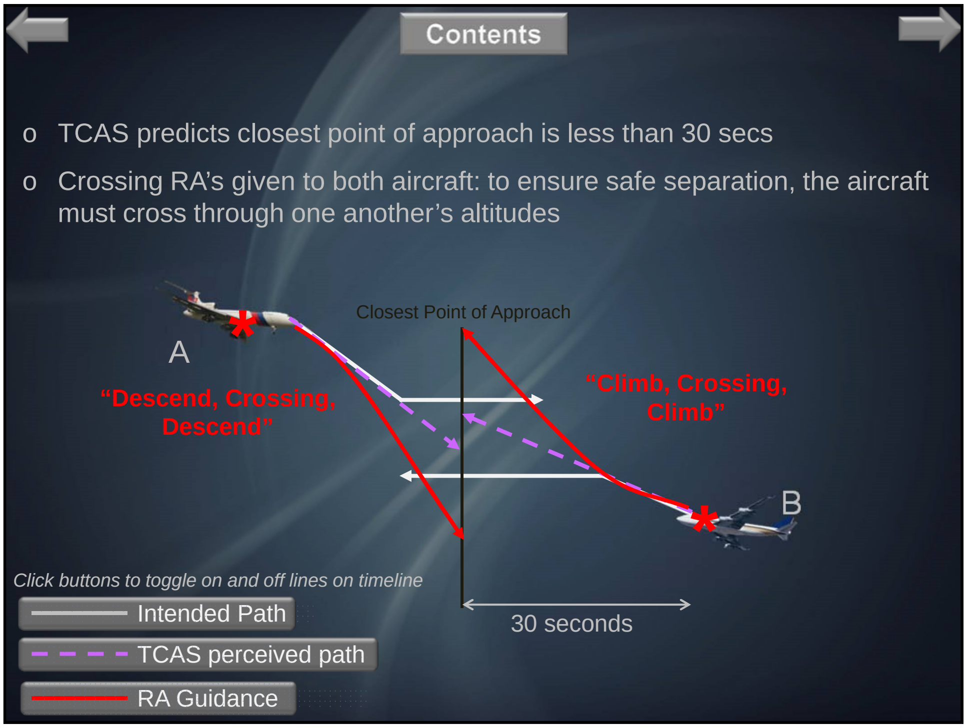


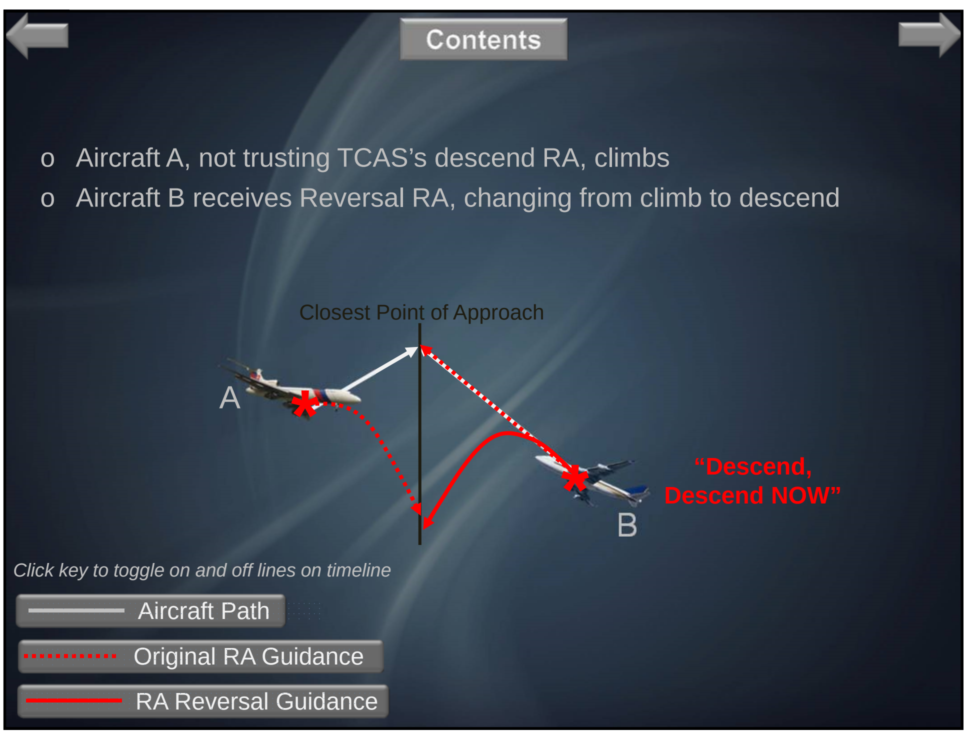

Contents
o TCAS predicts closest point of approach in less than 45 secs
o Both pilots receive a TA
A
Closest Point of Approach
"Traffic, Traffic"
"Traffic, Traffic"
45 seconds
B
Click buttons to toggle on and off lines on timeline
Intended Path
TCAS perceived path


Contents
o Aircraft A: Preventive RA, climb sense- i.e., do not descend
o Aircraft B: Weakening RA- i.e. reduce climb rate
o At clear of conflict, both aircraft have changed altitude by less than 500 feet
Closest Point of Approach
A
"Monitor Vertical Speed"
"Adjust Vertical Speed, Adjust"
B
Click buttons to toggle on and off lines on timeline
Intended Path
TCAS perceived path
RA Guidance


Contents
Timeline 2: Crossing and Reversal RA's
o Aircraft A: descending to cleared altitude
o Aircraft B: climbing to an altitude 1000 feet below A's cleared altitude
A
1000 feet
B
Intended Path


Contents
o TCAS predicts closest point of approach is less than 45 secs
o TA given to both pilots
A
"Traffic, Traffic"
Closest Point of Approach
B
45 seconds
"Traffic, Traffic"
Click buttons to toggle on and off lines on timeline
Intended Path
TCAS perceived path


Contents
o TCAS predicts closest point of approach is less than 30 secs
o Crossing RA's given to both aircraft: to ensure safe separation, the aircraft must cross through one another's altitudes
Closest Point of Approach
A
"Descend, Crossing, Descend"
"Climb, Crossing, Climb"
B
Click buttons to toggle on and off lines on timeline
30 seconds
Intended Path
TCAS perceived path
RA Guidance


Contents
o Aircraft A, not trusting TCAS's descend RA, climbs
o Aircraft B receives Reversal RA, changing from climb to descend
Closest Point of Approach
A
"Descend, Descend NOW"
B
Click key to toggle on and off lines on timeline
Aircraft Path
Original RA Guidance
RA Reversal Guidance

Contents

Mid-Training Quiz

Click options to see answer

What information can TCAS use to generate alerts?

a) Time until closest point of approach
b) Airspace classification
c) Vertical distance from the intruder
d) A and C
e) Not enough information

TCAS advisory logic assumes

a) I will perform a vertical maneuver in response to an RA (if needed)
b) I will follow the RA within 5 secs
c) Both A and B
d) TCAS logic doesn't make assumptions about my response

Contents

Displaying the RA

- Vertical maneuver guidance depicted on VSI, Attitude Indicator, or HUD
- For any depiction, fly out of the red

VSI:
Maintain a vertical speed greater than the red tape

Attitude Indicator:
Keep your nose outside the trapezoid

Contents

Excessive response to RA’s

TCAS assumes ¼ g maneuver in response to an RA

- Excessive response to an RA may cause significant disruption to air traffic operations and/or possible injury to passengers and crew
- Fly out of the red, but avoid an overly aggressive response
- By following the displayed vertical speed, altitude deviation should typically be no more than 300 to 500 feet

Contents

Abbreviated List of Reported Injuries due to Excessive TCAS Response

03/93	Continental Airlines B727	Houston, TX	1 injury
01/97	America West Airlines B737	Las Vegas, NV	1 injury
06/97	United Airlines B737	Valparaiso, IN	2 injuries
07/00	United Airlines B737	Chicago, IL	1 injury
01/01	Japan Airlines B747	Japan	99 injuries
06/01	Air Canada B737	Empress, Alberta	3 injuries
07/04	Comair Bombardier CL-600	Snow Hill, VA	1 injury
10/05	United Express Embraer 170	Dulles, VA	1 injury
11/06	Far Eastern Air B757	Jeju Island, Korea	26 injuries
10/09	AirTran Airways B717	Westchester, NY	1 injury

Contents

ATC Interaction

If given conflicting guidance from air traffic control and TCAS

- ✓ Follow TCAS guidance
- ✓ When able, report to ATC your inability to follow their instructions

Contents

Clear of Conflict

- o Clear of Conflict annunciated when a minimum vertical separation is ensured and/or total separation is increasing
- o Pilot should:
  - ✓ Re-engage autopilot and flight directors
  - ✓ Return to cleared altitude (if deviated)
  - ✓ Contact ATC and inform controller of the event

Contents

Mid-Training Quiz

Click options to see answer

To comply with a TCAS RA (active or crossing types)

a) Pitch the aircraft into the red and turn to avoid traffic
b) Pitch the aircraft out of the red; horizontal maneuvers are not necessary to avoid traffic
c) Call ATC and ask for permission to maneuver
d) None of the above

Maneuvering at the displayed vertical speed generally causes an altitude deviation of no more than _______

a) 500 feet
b) 750 feet
c) 1000 feet
d) 1200 feet
e) Not enough information

Contents

Mid-Training Quiz

Click options to see answer

Maneuvering well above the advised vertical speed given by TCAS

a) Is necessary to comply with the TCAS RA
b) Provides an added safety margin in separation assurance
c) Can cause injury to passengers and/or flight attendants
d) May alter air traffic flows and possibly put your aircraft in the path of other traffic
e) Options A and B
f) Options C and D

After “Clear of Conflict” is annunciated, I should

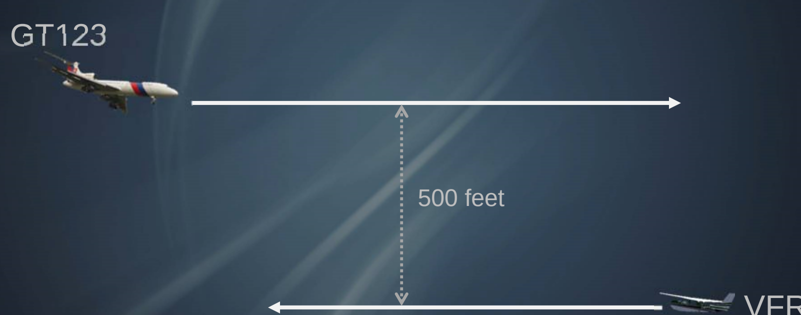
a) Inform ATC and ask for a new clearance at the altitude reached at “Clear of Conflict”
b) Not inform ATC, as they are likely aware of the RA using information provided by own guidance equipment
c) Inform ATC and return to the previously cleared altitude
d) Not enough information

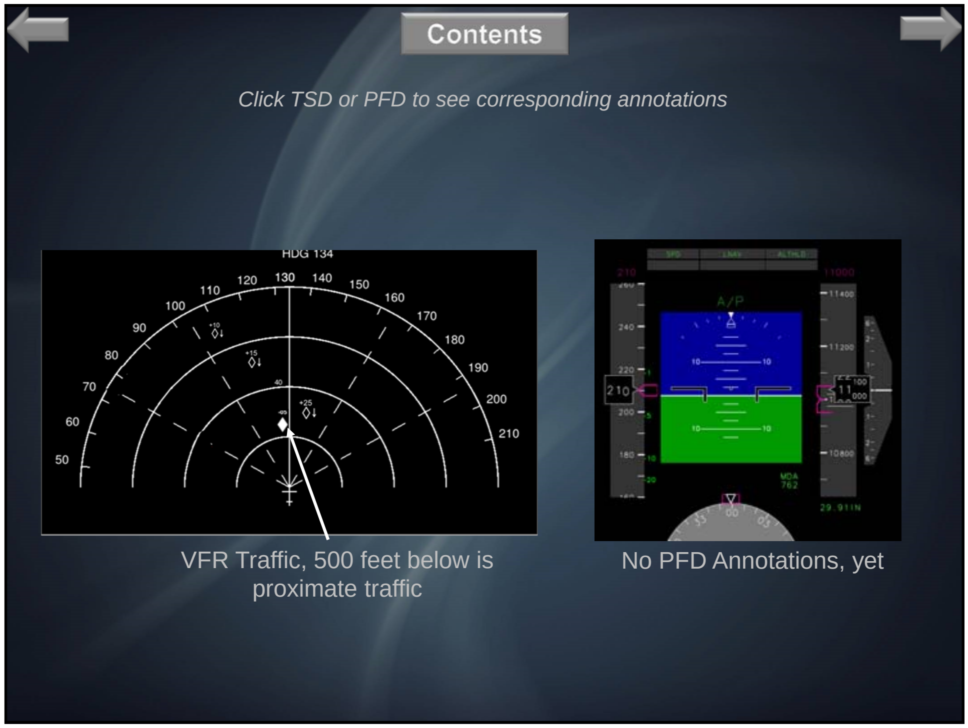
Contents

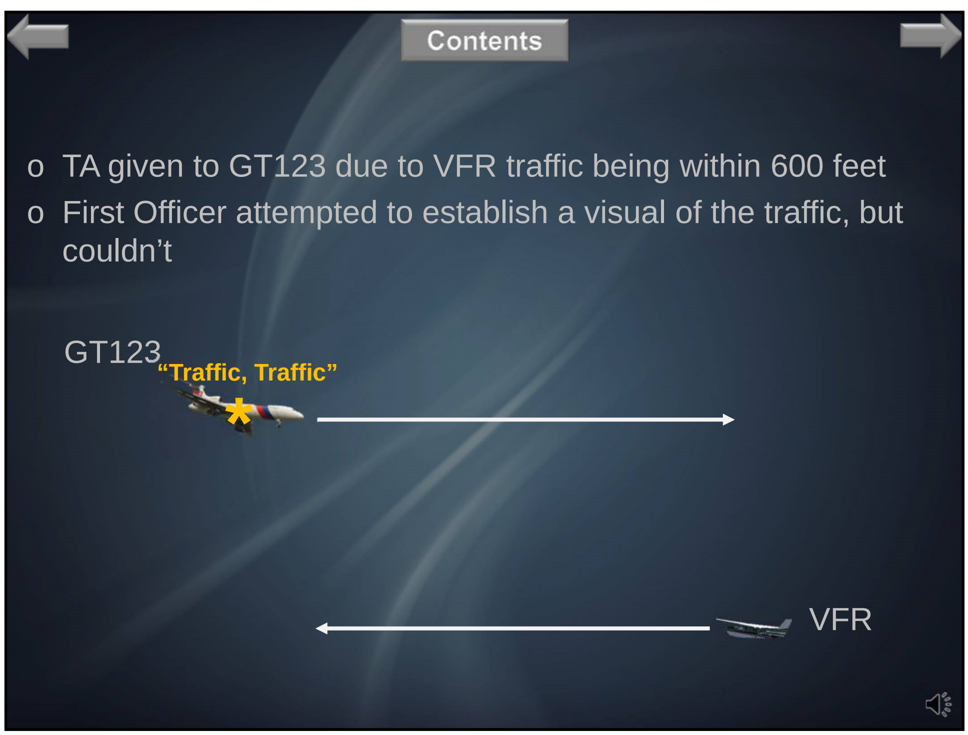
Example Timeline of a TCAS Event

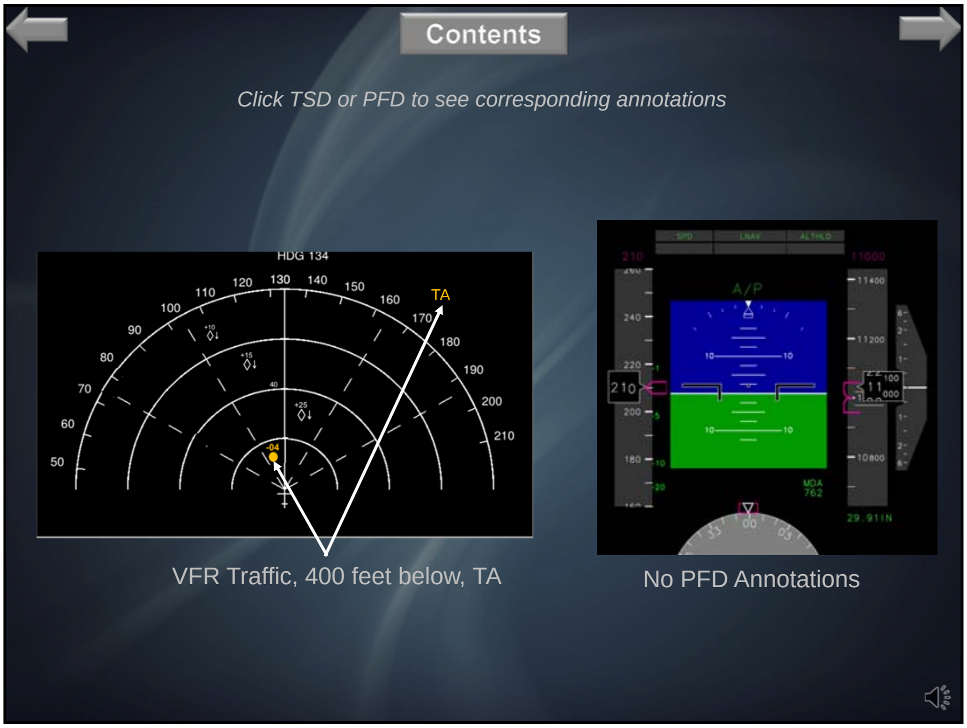
Contents

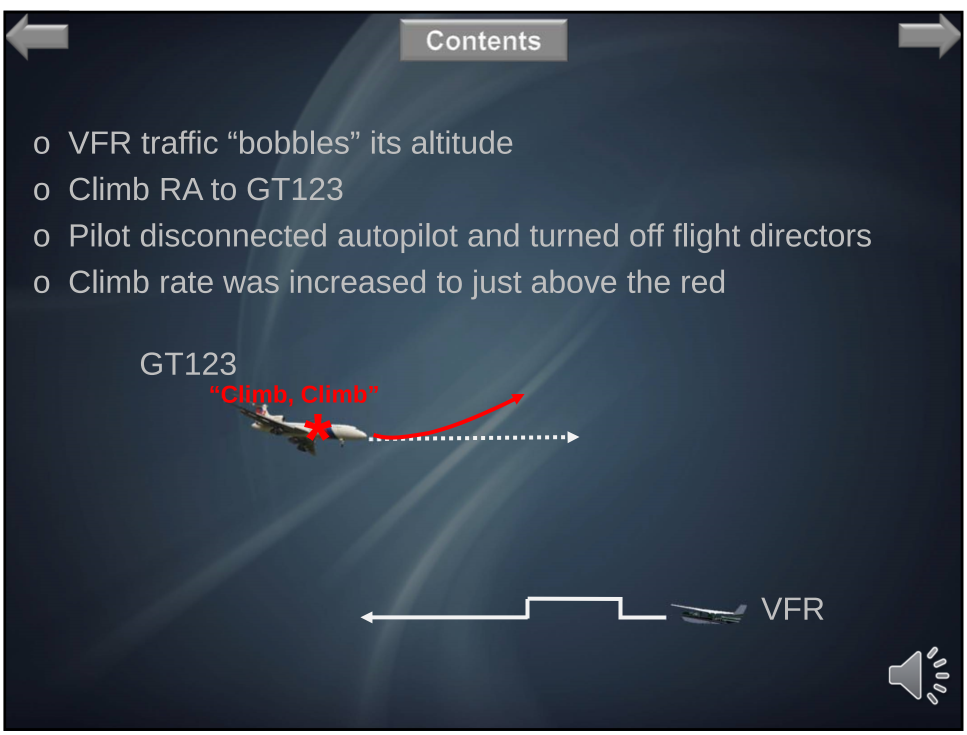
- ATC advises GT123 of VFR traffic passing 500 feet below

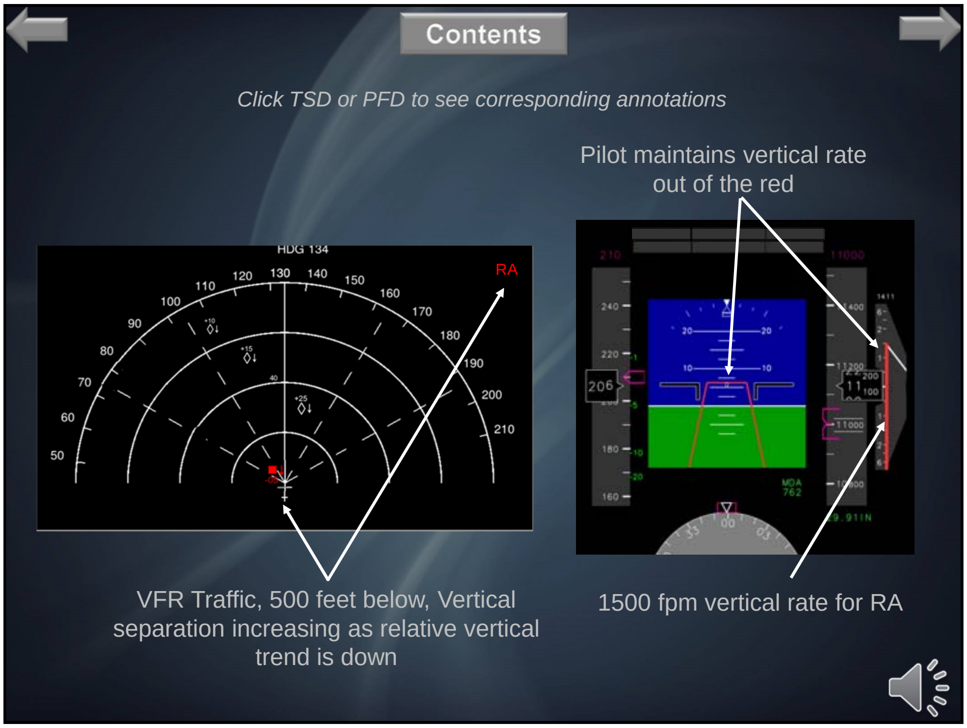


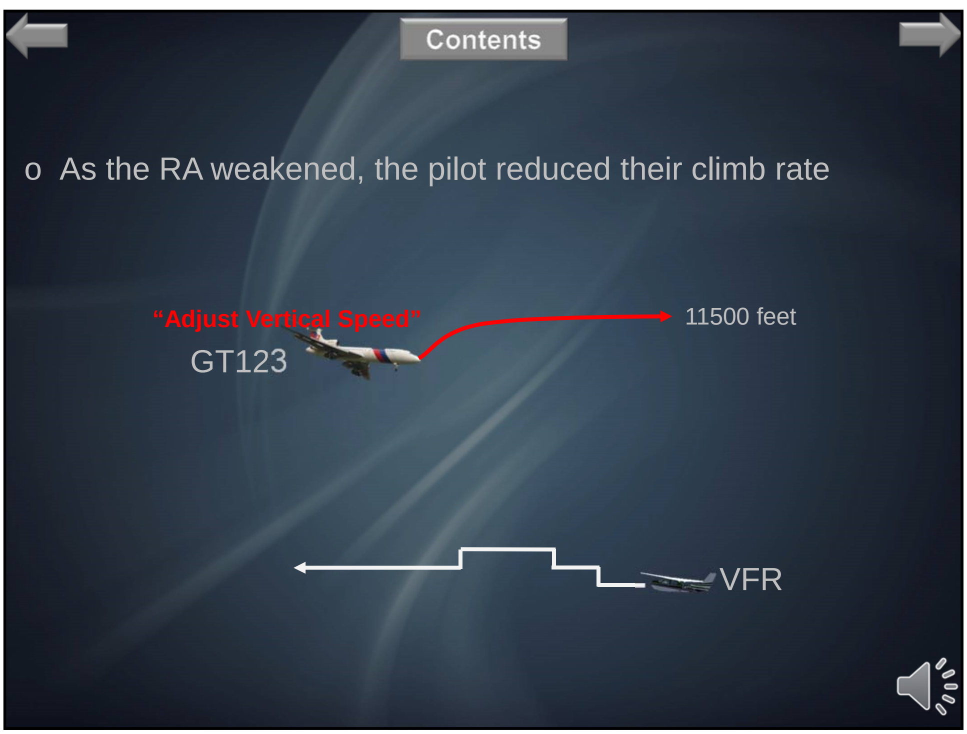

Contents
Click TSD or PFD to see corresponding annotations
HDG 134
VFR Traffic, 500 feet below is proximate traffic
No PFD Annotations, yet

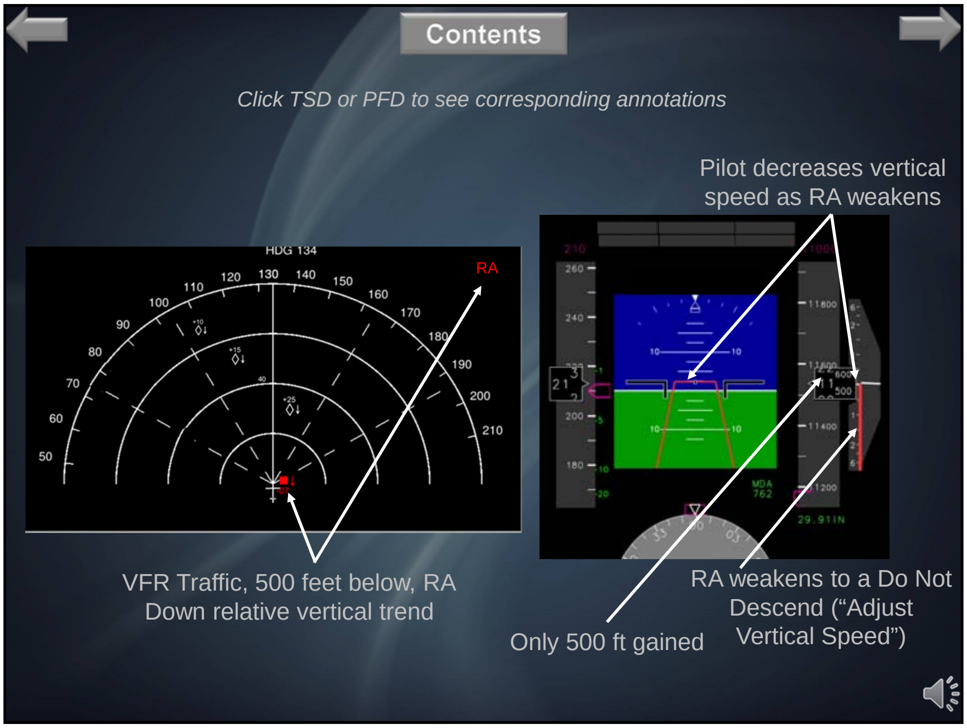

Contents
o TA given to GT123 due to VFR traffic being within 600 feet
o First Officer attempted to establish a visual of the traffic, but couldn't
GT123
"Traffic, Traffic"
VFR

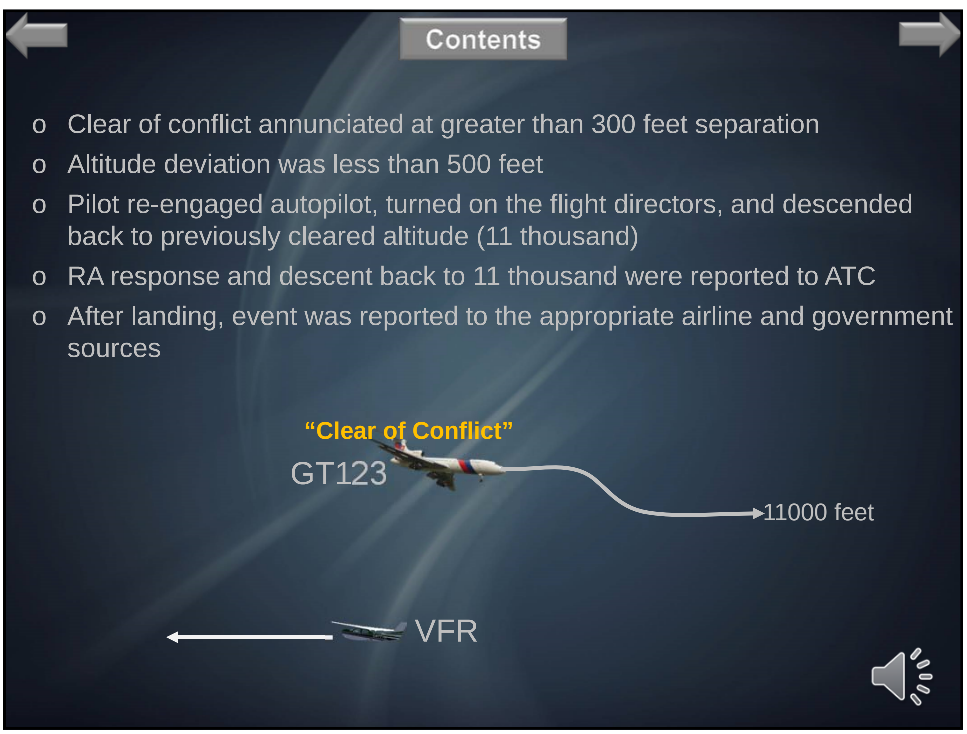

Contents
Click TSD or PFD to see corresponding annotations
HDG 134
TA
VFR Traffic, 400 feet below, TA
No PFD Annotations

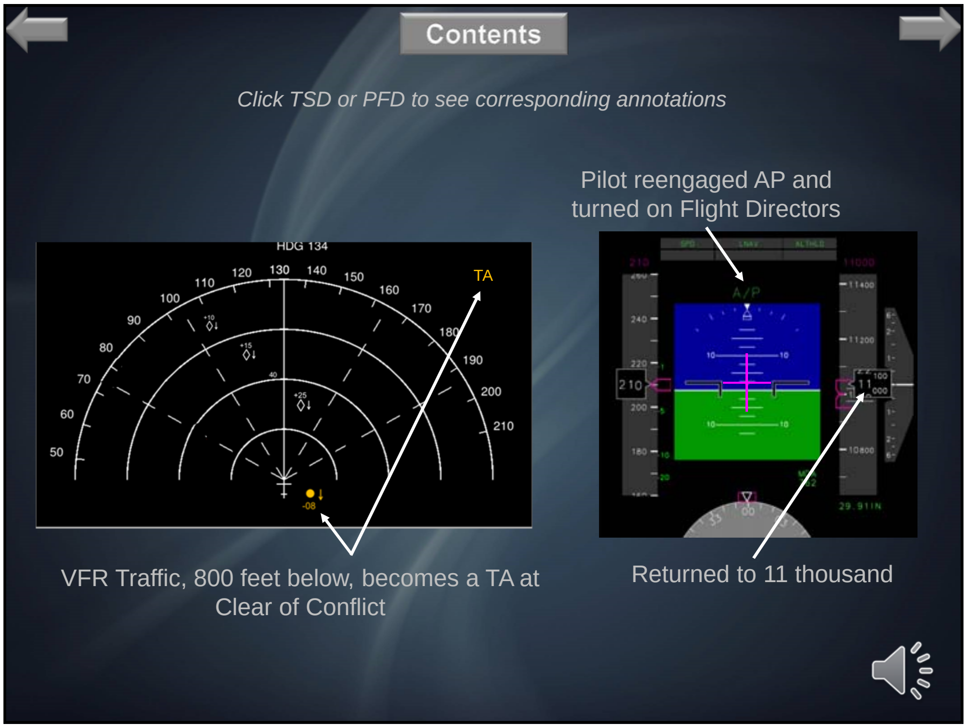

Contents
o VFR traffic “bobbles” its altitude
o Climb RA to GT123
o Pilot disconnected autopilot and turned off flight directors
o Climb rate was increased to just above the red
GT123
“Climb, Climb”
VFR


Contents
Click TSD or PFD to see corresponding annotations
Pilot maintains vertical rate out of the red
HDG 134
RA
206
VFR Traffic, 500 feet below, Vertical separation increasing as relative vertical trend is down
1500 fpm vertical rate for RA


Contents
o As the RA weakened, the pilot reduced their climb rate
"Adjust Vertical Speed"
GT123
11500 feet
VFR


Contents
Click TSD or PFD to see corresponding annotations
Pilot decreases vertical speed as RA weakens
HDG 134
RA
VFR Traffic, 500 feet below, RA Down relative vertical trend
Only 500 ft gained
RA weakens to a Do Not Descend ("Adjust Vertical Speed")


Contents
o Clear of conflict annunciated at greater than 300 feet separation
o Altitude deviation was less than 500 feet
o Pilot re-engaged autopilot, turned on the flight directors, and descended back to previously cleared altitude (11 thousand)
o RA response and descent back to 11 thousand were reported to ATC
o After landing, event was reported to the appropriate airline and government sources
"Clear of Conflict"
GT123
11000 feet
VFR


Contents
Click TSD or PFD to see corresponding annotations
Pilot reengaged AP and
turned on Flight Directors
HDG 134
TA
VFR Traffic, 800 feet below, becomes a TA at
Clear of Conflict
Returned to 11 thousand


Contents
Congratulations!
You have completed the TCAS
training program!
Start Simulator
Training Scenarios
Return to
Contents

TCAS: Simulator Training

Training Event 1

Please notify your instructor when you are ready to proceed

Training Event 1

TCAS Event Type: Descend RA

Did you:
- ✓ Disengage autopilot and flight directors?
- ✓ Maneuver your aircraft out of the red?
- ✓ Ensure your descent rate was not excessive?
- ✓ Decrease descent rate for weakening RA's?
- ✓ (if needed) increase descent rate for strengthening RA's?

After Clear of Conflict, did you:
- ✓ Reengage autopilot and flight directors?
- ✓ Return to cleared altitude if needed?
- ✓ Notify ATC of the event if clearance was violated?

Read more

5 minutes added to time

Continue to Next Training Event

Training Event 2

Please notify your instructor when you are ready to proceed

Training Event 2

TCAS Event Type: Climb RA

Did you:

- ✓ Disengage autopilot and flight directors?
- ✓ Maneuver your aircraft out of the red?
- ✓ Ensure your climb rate was not excessive?
- ✓ Decrease climb rate for weakening RA's?
- ✓ (if needed) increase climb rate for strengthening RA's?

After Clear of Conflict, did you:

- ✓ Reengage autopilot and flight directors?
- ✓ Return to previously cleared altitude?
- ✓ Notify ATC of the event?

Read more

5 minutes added to time

Continue to Next Training Event

Training Event 3

Please notify your instructor when you are ready to proceed

Training Event 3

TCAS Event Type: Crossing Descend RA

Did you:

- ✓ Disengage autopilot and flight directors?
- ✓ Maneuver your aircraft out of the red?
- ✓ Ensure your descent rate is not excessive?
- ✓ Decrease descent rate for weakening RA's?
- ✓ (if needed) increase descent rate for strengthening RA's?

After Clear of Conflict, did you:

- ✓ Reengage autopilot and flight directors?
- ✓ Attempt to re-establish the previously cleared flight path?
- ✓ Notify ATC of the event?

Read more

5 minutes added to time

Continue to Next Training Event

Training Event 4

Please notify your instructor when you are ready to proceed

Training Event 4

TCAS Event Type: Climb RA
Conflicting ATC Instructions

Did you:

- ✓ Follow the RA?
- ✓ If able, notify ATC of your inability to comply with their directions?
- ✓ Disengage autopilot and flight directors?
- ✓ Maneuver your aircraft out of the red?
- ✓ Ensure your climb rate is not excessive?

After Clear of Conflict, did you:

- ✓ Reengage autopilot and flight directors?
- ✓ Attempt to re-establish the cleared flight path?
- ✓ Notify ATC of the event?

Read more

5 minutes added to time

Continue to Next Training Event

Training Event 5

Please notify your instructor when you are ready to proceed

Training Event 5

TCAS Event Type: Preventive RA
VFR Traffic Passing 500 Feet Below

Did you:
- Keep your aircraft out of the red?

After Clear of Conflict, did you:
- Reengage autopilot and flight directors, if disengaged?
- Notify ATC of the event?

Read more

5 minutes added to time

Finish Training

Congratulations!

You have completed the TCAS simulator training!

If you have any questions regarding this training session, please ask the instructor at this time.

APPENDIX B: EXPERIMENT AND PILOT DOCUMENTS

Pilot briefing

The purpose of this experiment is to investigate the interaction between you, ATC, and TCAS in realistic operations, after having completed a revised training program for TCAS. We would like to see your most natural response during the data portion of this experiment.

Overview

You will fly a fixed-base, desktop flight simulator of the Boeing 747-400. During every flight you will sit in the left seat and act as Captain and as Pilot Flying (PF). You will be assisted by a First Officer (FO) who is fresh out of training – he or she knows the aircraft systems well, but has little operational experience. He/she will act as pilot managing (PM) and his/her main tasks will be to handle air traffic communications and to assist you with the checklists, providing any inputs to the Flight Management System (FMS) or Mode Control Panel (MCP) that you request, or any other tasks that you command. Your FO is here to help you and will not deliberately do anything wrong.

The B747-400 simulator is shown in Figure 1. The sidestick on your left is yours to fly the aircraft- the red trigger button disconnects the autopilot. Eye motion will be tracked using infrared by a FaceLAB eye tracker. It is mounted below the PFD and you do not need to wear anything on your head. This device is not intended to distract you and will not affect your vision. The eyetracker records which display you are looking at, but doesn’t record video. Before your simulation runs, we will calibrate the eye tracker through a few simple calibration tasks, such as looking into either the left or right lens.

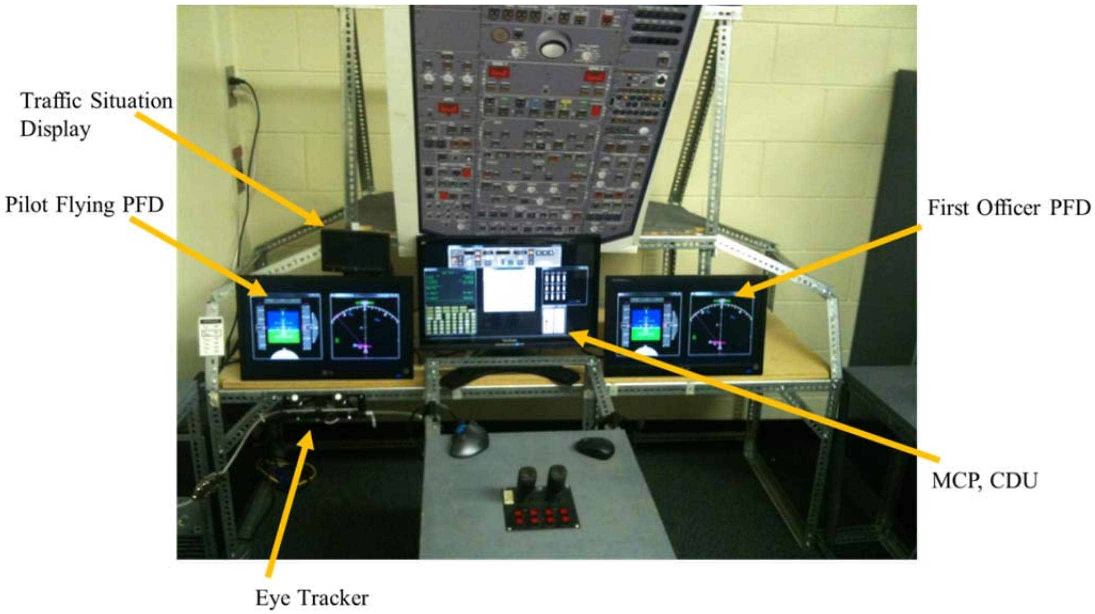


FIGURE 1. Image of the Flight Deck

Flight tasks

Your task is to fly a Standard Arrival Route (STAR), including performing the approach checklist at the start of the flight. Typically, the flight will start around an altitude of 10,000 to 20,000 feet and last 15 minutes. Your FMS has been pre-programmed with your approach, arrival and the transition between them. The aircraft will be started in a clean configuration (trimmed, no speed brakes, zero flaps, gear up). The wind will be calm, but Instrument Meteorological Conditions (IMC) apply, and there will be no visibility out the window. The flight generally ends once the localizer is within one dot.

The air traffic controller will be controlling you and other aircraft in the vicinity of the airport, and you will hear the party-line in the background. As Pilot-Managing your FO will be responsible for ATC communications, but you are welcome to request any communications from your FO (or step in and speak directly to ATC) as you would in actual operations. Your call sign to ATC will be GT123.

During your flight, you may encounter some events requiring action. There is no right-or-wrong response – we are interested in your best judgment of what you think the situation calls for based on the information you are given at that time. This includes not only how you fly the aircraft and manage the aircraft systems with the help of your FO, but also how you interact with ATC. Please try to act as you would in the real aircraft, with passengers and when flying in IMC, so that our study can indicate how TCAS can best be improved to support you.

Simulator operation

Here are the main elements of the Primary Flight Display (PFD):

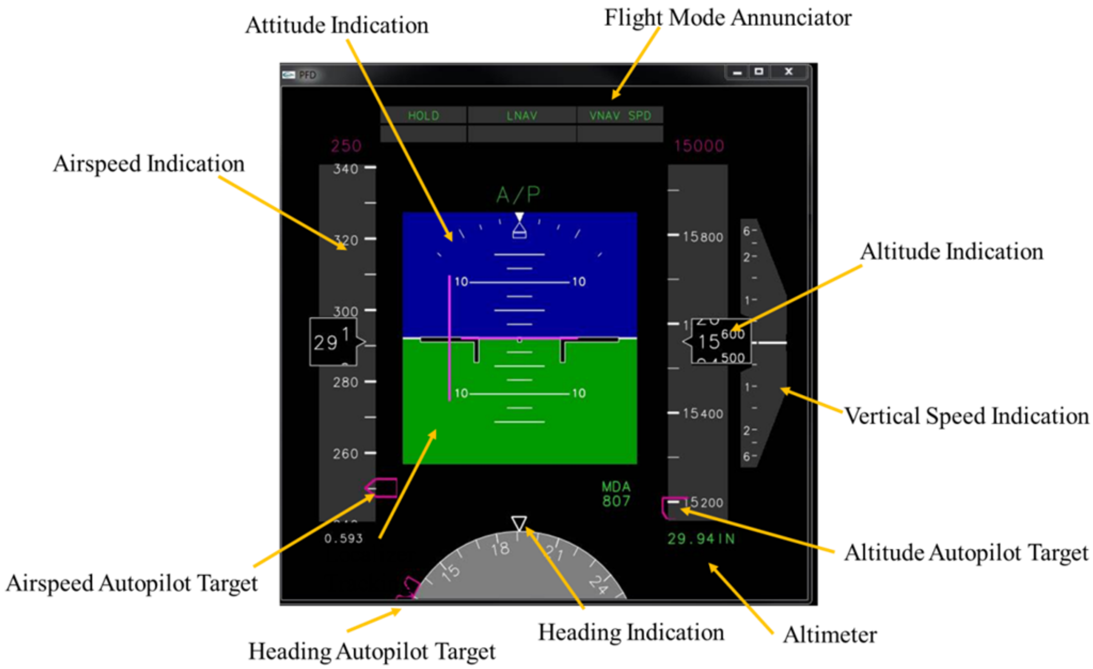


Figure 2. Primary Flight Display

For the purposes of this exercise, you will be using the MAP mode of the Navigation Display (ND). In MAP mode the ND is oriented with respect to the aircraft's current heading. You can change its range and toggle stations, waypoints and airports using your ND control panel, or ask your FO to do it.

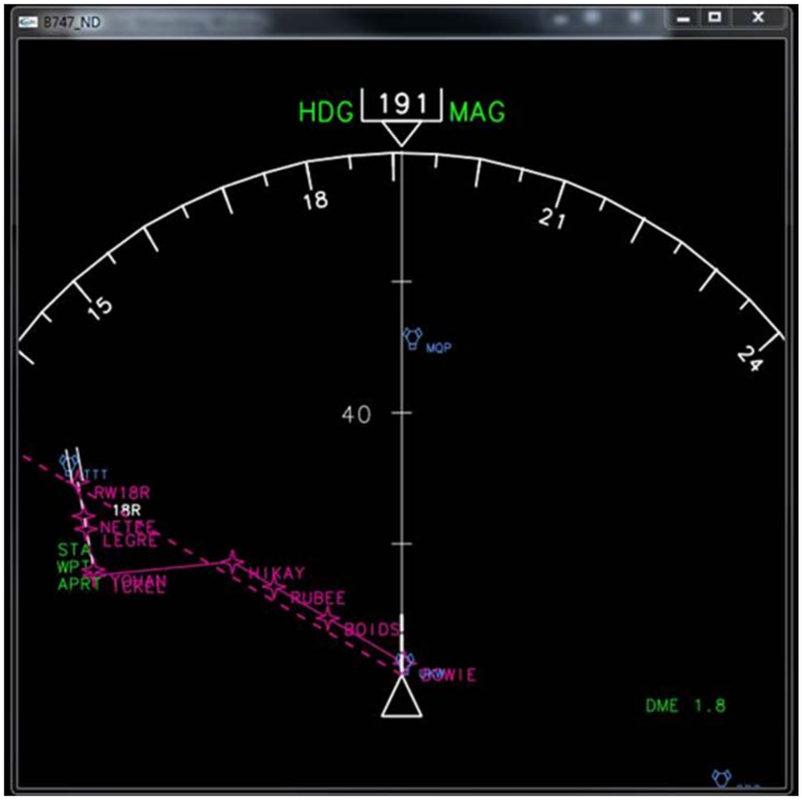


Figure 3. Navigation Display

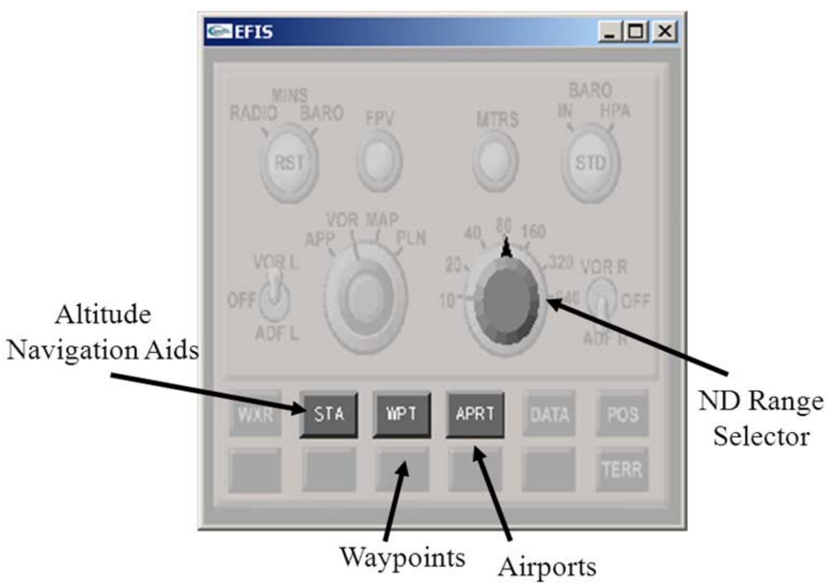


Figure 4. ND Control Panel

In all flights you will start with the autopilot engaged. Feel free to disengage and re-engage the autopilot as you would in real-life when a traffic event is encountered. Flight director guidance will continue to be shown to you when the autopilot disengages, unless you also choose to disengage the flight director, or ask your FO to disengage it for you. The throttles have been set to auto-throttle, and do not need any adjustment to control aircraft altitude or speed.

Localizer and Approach modes can be armed by clicking “LOC” and “APP” respectively. This will bring up the localizer and glideslope indicators in your PFD. The simulator is not capable of engaging these modes and tracking the ILS; we will generally terminate the flight when you get within one dot on the localizer.

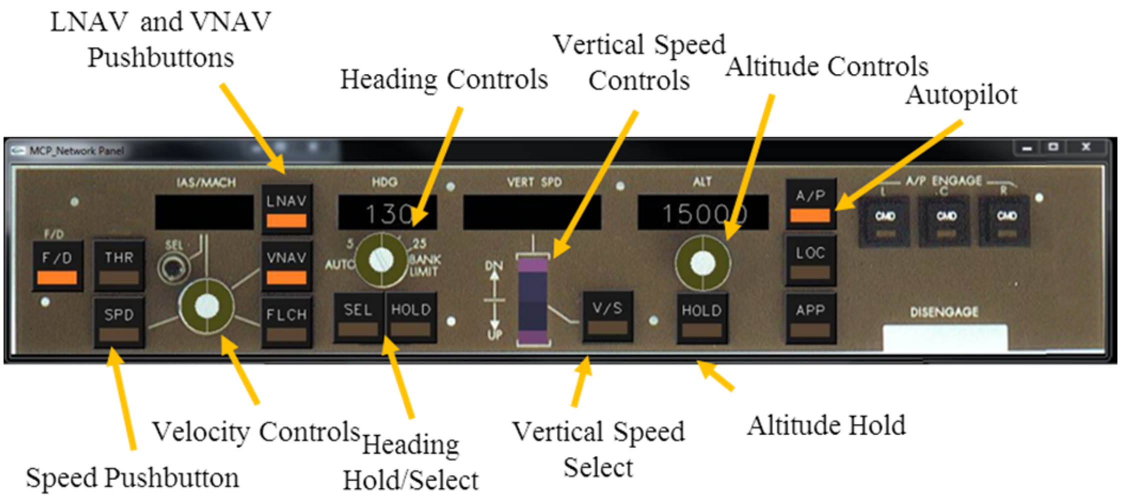


Figure 5. MCP Panel

Pre-Experiment Questionnaire

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I understand completely how TCAS operates					
I know when to expect TCAS advisories					
I trust TCAS advisories					
I follow the maneuvers displayed by TCAS advisories					
I understand TCAS maneuvers when they are issued					
TCAS is a useful tool in the cockpit					
TCAS is unnecessarily distracting					
TCAS is necessary for safe operation					
I can maintain safe separation without TCAS but with					
I reference the TCAS traffic situation display during					
I reference the TCAS traffic situation display after a TA					
I reference the TCAS traffic situation display after an RA					

What are the current procedures for following a TCAS RA with your airline?

What type of TCAS training have you completed?

_____ Videos/reading material ______ Classroom _______ Simulator based

Do you feel your previous TCAS training has prepared you for TCAS events? If not, why?

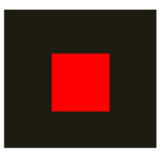
Pre-training TCAS Quiz

1) What information can TCAS use to generate alerts?
   a) The horizontal distance from intruder
   b) The time until closest point of approach
   c) The intruder's aircraft type
   d) Both A and B
   e) Both A and C
   f) All of the above

2) TCAS advisory logic assumes
   a) I will follow the RA's vertical maneuver guidance within 5 seconds of receiving the advisory
   b) I will perform a vertical maneuver in response to an RA
   c) Both A and B
   d) TCAS logic does not make assumptions about my response to advisories

3) Please match the following symbols with the type of traffic corresponding to the symbol

a)



1. Traffic Advisory

b)

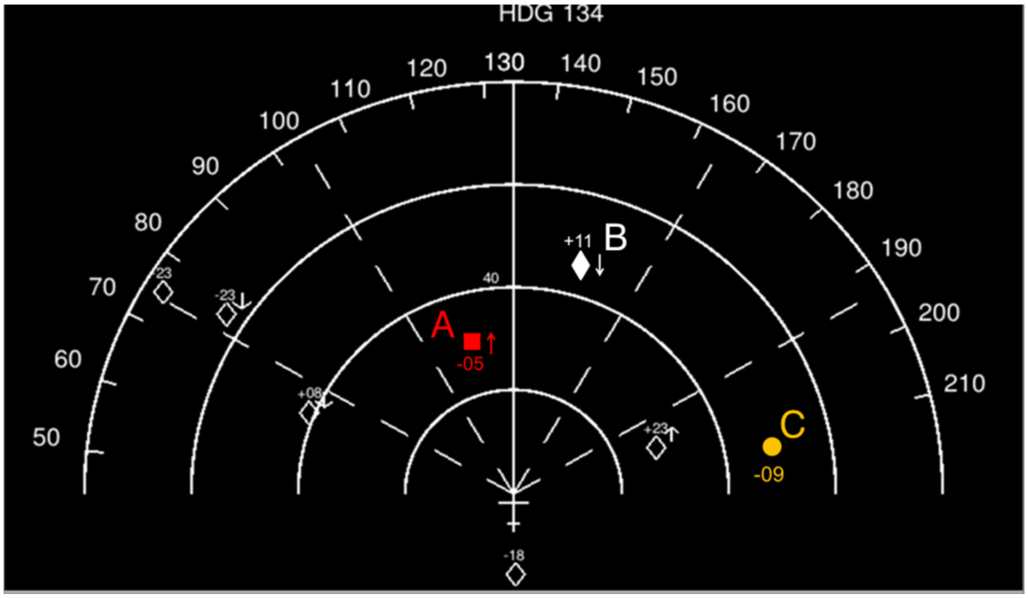
2. Proximate Traffic

c)

3. Resolution Advisory

d)

4. Other traffic (no factor)



Please answer questions 4-6 using the image shown below

4) If my aircraft is level at 11,000 feet, symbol "B" is flying at what altitude?
   a) 1,100 feet
   b) 9,900 feet
   c) 10,500 feet
   d) 11,000 feet
   e) 12,100 feet
   f) Not enough information provided

5) Assuming my aircraft is level, in what vertical direction is symbol "C" traveling?
   a) Climbing
   b) Descending
   c) Level with less than 500 feet per minute climb or descend
   d) Not enough information provided

6) Using only the information provided on this display, I would
   a) Turn
   b) Climb
   c) Descend
   d) Not enough information provided

7) To comply with a TCAS RA, I would
   a) Pitch the aircraft out of the red area and perform a turn away from traffic
   b) Pitch the aircraft out of the red area

c) Call ATC and ask for permission to maneuver.
d) Not enough information provided

8) Typically, following a corrective RA will cause a deviation in my altitude of no more than
a) 500 feet
b) 1000 feet
c) 1500 feet
d) 2000 feet
e) Depends on the RA type
f) Not enough information provided

9) If maneuvering to a TCAS RA causes a deviation from my ATC clearance, I should
a) Inform ATC of the maneuver and ask for a new clearance at the altitude established after "Clear of Conflict"
b) Inform ATC of the maneuver and return to the originally cleared altitude
c) Not inform ATC, as they are likely aware of the RA using information provided by own guidance equipment
d) Listen to see if the other aircraft also received a conflict
e) Not enough information provided

10) Maneuvering well above the advised vertical speed given by TCAS
a) Is necessary to comply with the TCAS RA
b) Can cause injury to passengers and/or flight attendants
c) Provides an added safety margin in separation assurance
d) May alter air traffic flows and possibly put your aircraft in the path of other traffic
e) Options A and C
f) Options B and D

11) Assuming I am acting as pilot flying, if ATC and a TCAS RA give conflicting maneuver instructions, I should follow the instructions given by
a) ATC
b) TCAS
c) My pilot monitoring
d) Not enough information provided
e) None of the above

Scenario Descriptions *(given to each pilot before each flight)*

Tutorial flight #1

This flight will start you on the CEDAR CREEK Six STAR just past BELLS approaching TACKE. Your aircraft will start at an altitude of 11,000 feet with an indicated airspeed of 287 kts slowing to 250 and a heading of 311 degrees. You have been told to expect runway 17C.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

Tutorial flight #2

This flight will start you on the BOWIE Niner STAR just beyond ACKME approaching UKW (BOWIE). Your aircraft will start at an altitude of 18,000 feet with an indicated airspeed of 298 kts slowing to 290 kts and a heading of 205 degrees. You have been told to expect runway 18R.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

Training Flight Descriptions

TT1

You will start on the BOWIE Niner STAR at UKW (BOWIE). Your aircraft will start at an altitude of 18,000 feet descending to 15,000 feet with an indicated airspeed of 285 kts and a heading of 123.

TT2

You will start on the CEDAR CREEK Six STAR past Dietz. Your aircraft will start at an altitude of 11,000 feet with an indicated airspeed of 210 kts and a heading of 0 degrees.

TT3

You will start on the BONHAM Five STAR just beyond KARLA. Your aircraft will start at an altitude of 11,000 feet with an indicated airspeed of 245 kts slowing to 210 kts and a heading of 221 degrees.

TT4

You will start on the BONHAM Five STAR approaching BYP. Your aircraft will start at an altitude of 15,000 feet with an indicated airspeed of 250 kts and a heading of 230 degrees.

TT5

You will start on the BONHAM Five STAR prior to LEMYN. Your aircraft will start at an altitude of 7,000 feet with an indicated airspeed of 223 kts slowing to 210 kts and a heading of 230 degrees.

A

You will start on the BOWIE Niner STAR at UKW (BOWIE). Your aircraft will start at an altitude of 15,600 feet descending to 15,000 feet with an indicated airspeed of 291 kts slowing to 250 kts and a heading of 191 degrees turning to the next course leg at 130 degrees. You have been told to expect runway 18R.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

B

You will start on the BOWIE Niner STAR just beyond PLEBS approaching UKW (BOWIE). Your aircraft will start at an altitude of 18,000 feet descending to 15,000 feet with an indicated airspeed of 285 kts and a heading of 122 degrees. You have been told to expect runway 18R.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

C

You will start on the CEDAR CREEK Six STAR just beyond BELLS approaching TACKE. Your aircraft will start at an altitude of 11,000 feet with an indicated airspeed of 287 kts slowing to 280 kts and a heading of 311 degrees. You have been told to expect runway 17C.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

D

You will start on the BONHAM Five STAR just beyond ROBEY approaching BYP (BONHAM). Your aircraft will start at an altitude of 17,000 feet descending to 15,000 feet with an indicated airspeed of 280 kts and a heading of 247 degrees. You have been told to expect runway 17L.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

E

You will start on the BOWIE Niner STAR just beyond PLEBS approaching UKW (BOWIE). Your aircraft will start at an altitude of 15,600 feet descending to 15,000 feet with an indicated airspeed of 290 kts slowing to 250 kts and a heading of 190 degrees turning to 130 degrees. You have been told to expect runway 18R.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

F

You will start on the CEDAR CREEK Six STAR just beyond BELLS approaching TACKE. Your aircraft will start at an altitude of 11,000 feet with an indicated airspeed of 287 kts slowing to 250 kts and a heading of 311 degrees. You have been told to expect runway 17C.

Both your STAR and approach paths have already been programed into the FMS for you. Your path will be visible on the ND. Working with the FO, your tasks for the tutorial are the following:

1. Please take a moment to orient yourself
2. Notify the researcher when you are ready to begin the flight.
3. Fly the arrival and approach intercept that have already been programmed into the FMS as you would on a real flight, including responding to air traffic or TCAS events.
4. Perform the approach briefing at the start of the flight
5. The flight will end once you are getting established on approach.

If at some point you experience any difficulty with the simulator, please try to resolve the problem in a safe and professional way, as you would in real-life. This includes delegating systems tasks to your First Officer as appropriate and contacting ATC when necessary.

Post-Run Questionnaire

Which maneuver did the TCAS RA command? ________________________________

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I agreed with the maneuver displayed by the RA					
I followed the maneuver displayed by the RA					
I understood why TCAS commanded this maneuver					
I trusted the TCAS maneuver					
I knew when to maneuver					
I knew the best maneuver to perform					
The TCAS TA happened too soon.					
The TCAS TA happened too late					
The TCAS RA happened too soon.					
The TCAS RA happened too late.					
The traffic event was the result of ATC's action or inaction					
The traffic event was the result of another pilot's actions					
The TCAS RA was a necessary advisory					
The TCAS RA conflicted with ATC instructions					

What information sources did you see/hear, and what information determined your actions?

	Saw/heard	Determined Actions (Describe)
ATC call out of traffic to you		
ATC instructions you		
Information from your FO		
Party Line Information		
TCAS traffic situation display		
TCAS traffic advisory		
TCAS resolution advisory		
Other:		

Post-Experiment Questionnaire

Age (circle): <30   30-39   40-49   50-59   60-65   Gender (circle):   Male   Female

Rank (circle): Captain   First Officer   Initial training: Military   Civilian

Total # of Flight Hours: _____________

Aircraft Current In ____

Hours in Current Type ____

Prior Glass Aircraft ____

On scale of 1-5, rate familiarity with the simulated airspace (5 being very familiar)
____

Do you have any experience with dogfighting / air combat? Describe.

____

____

Please describe the TCAS implementation you are most familiar with:

RA Maneuver shown on: PFD as 'fly-to' bars   PFD – vertical speed

Separate TCAS display   VSI

Traffic shown on: ND   Separate display (without RA VSI)

Separate TCAS display (with RA VSI)

Up to what range is traffic shown on your traffic display? __________________

Are you accustomed to any other presentation of traffic on the flightdeck (e.g. ADS-B)? If so, how do you use it for collision avoidance / separation assurance?

Assessing the TCAS Training Program

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The computer based portion of the training program was easy to understand					
The computer based portion of the training program was easy to use					
The computer based portion of the training program taught confusing material in an easy to understand method					
I liked the computer based portion of the training program					
The simulator based portion of the training program was easy to understand					
The simulator based portion of the training program was easy to use					
The simulator based portion of the training program taught confusing material in an easy to understand method					
I liked the simulator based portion of the training program					
Integrating Computer Based and Simulator Based Training aided my understanding of TCAS					
I learned new information about TCAS from today's training					
My understanding of TCAS has increased					
I am more likely to comply with RA's after completing the training program					
I am less likely to perform excessive maneuvers in response to RA's after completing today's training					
I am more likely to trust TCAS after completing today's training					
Today’s training program was very similar to the training program I have completed previously.					
The training program took too long to complete					

What would you say you learned today? ____

Do you have any other comments regarding the training program?

____

____

____

____

Was there any material taught today not originally included in previous classroom and/or simulator TCAS training?

____

____

____

____

(If you clicked "Read More" during the simulator training events)
What additional information would you have liked to see?

____

____

____

____

Assessing the Flight Simulation

Did you see any patterns in the traffic events?

____

____

____

Did the simulator allow you to respond as you would in real-life? If not, please describe how you might have reacted differently on a revenue flight.

____

____

____

Would you have responded differently if you had been in VMC?

____

____

APPENDIX C: RFS OUTPUTS

Recorded Value	Units
Current Time	sec
Latitude	deg
Longitude	deg
Altitude	ft
XPositionFESF	ft
YPositionFESF	ft
ZPositionFESF	ft
Euler Angle Theta	deg
Euler Angle Phi	deg
Euler Angle Psi	deg
Psi Dot	deg/sec
Phi Dot	deg/sec
Theta Dot	deg/sec
Beta Dot	deg/sec
Velocity U	fps
Velocity V	fps
Velocity W	fps
Ground Speed	knots
Indicated Airspeed	knots
True Airspeed	knots
Vertical Speed	fps
Mach	
Baro Altitude	ft
Radar Altitude	ft
Angle if Attach	deg
Sideslip	deg
Pitch	deg

Roll	deg
Flight Path Angle	deg
True Heading	deg
Magnetic Heading	deg
True Track	deg
Magnetic Track	deg
Bank Angle Limits	deg
Total Weight	lb
Total Fuel Weight	lb
Selected Fuel Tanks	
Flap Deflection	deg
Commanded Flap Deflection	deg
Maximum Propulsive Thrust	lb
Total Propulsive Thrust	lb
Altitude Commanded	ft
Heading Commanded	deg
Airspeed Commanded	kts
Vertical Speed Commanded	fps
Control Stick X	
Control Stick Y	
Rudder Control	
is Autopilot Engaged	
is Flight Director Engaged	
Speed Dial	
HDG Dial	
Vert Dial	
Alt Dial	
Speed Dial Right	
Speed Dial Left	
Heading Dial Right	
Heading Dial Left	

Vertical Dial Down	
Vertical Dial Up	
Altitude Dial Right	
Altitude Dial Left	
Localizer	
Approach	
Vertical Speed Mode	
Gear Commanded Positions	
Rudder Deflection	deg
Aileron Deflection	deg
Elevator Deflection	deg
Pitch Trim	deg
TCAS Advisory	
TCAS Combined Control	
TCAS Vertical Control	
TCAS Crossing	
TCAS Up Advisory	
TCAS Down Advisory	
TCAS Vertical Speed	
Strength	
Autopilot Mode	

APPENDIX D: GENERALIZED DATA

Data (Units/Comments)
Pilot Number
Run Number
Scenario Number
Event Number (9or 10=unexpected event)
Scenario (9 or 10 indicates unexpected event))
ATC Interaction (Alphabetic)
ATC Interaction (1= None, 2=Partyline, 3=Callout, 4=Conflicting)
Traffic Density
Expected RA Type (Alphabetic)
Expected RA Type (-1=TA, 1=Climb, 2=Descend, 3=Crossing Descend, 5=Preventive)
Actual RA Type
Did Not Match Expected
Intercept Approach
VFR
Pre TA Turn Autopilot Off (yes/no)
Pre TA Turn Auto Pilot Off (secs)
Pre TA Call ATC (yes/no)
Pre TA Call ATC (secs)
Pre TA- Duration Pilot Viewed ND (secs)
Pre TA- Duration Pilot Viewed TSD (secs)
Pre TA- Duration Pilot Viewed PFD (secs)
Pre TA- Duration Pilot Viewed CDU (secs)
Pre TA- Duration Pilot Viewed MCP (secs)
Autopilot Vertical Mode before TA (numeric)
Autopilot Lateral Mode before TA (numeric)
TA Turn Autopilot Off (yes/no)
TA Turn Autopilot Off (time after TA)

TA Call ATC (yes/no)

TA Call ATC (time after TA)

TA Call ATC (yes/no) (alternate)

TA- Duration Pilot Viewed ND (secs)

TA- Duration Pilot Viewed TSD (secs)

TA- Duration Pilot Viewed PFD (secs)

TA- Duration Pilot Viewed CDU (secs)

TA- Duration Pilot Viewed MCP (secs)

Index of TA Start

Total duration of TA (secs)

Aircraft Vertical Speed Just Before RA (fps)

Last Autopilot Vertical Mode before RA (numeric)

Last Autopilot Lateral Mode before RA (numeric)

RA Turn Autopilot Off (yes/no)

RA Turn Autopilot Off (secs)

Pilot calls ATC after RA (yes/no)

Pilot Calls ATC after RA (secs)

Did the pilot perform any large turns? (yes/no)

Amount of heading changed over duration of ra (degrees)

Pilot was in compliance for entire duration of RA with no buffer (yes/no)

Pilot was in compliance for entire duration of RA with 1/2 fps buffer(yes/no)

Pilot was in compliance for entire duration of RA with 1 fps buffer(yes/no)

Pilot was in compliance for entire duration of RA with 2 fps buffer(yes/no)

Pilot was in compliance for entire duration of RA with 3 fps buffer(yes/no)

Time after RA when pilot first matched RA rate with no buffer (seconds)

Time after RA when pilot first got within 1/2 fps of RA rate (seconds)

Time after RA when pilot first got within 1 fps of RA rate (seconds)

Time after RA when pilot first got within 2 fps of RA rate (seconds)

Time after RA when pilot first got within 3 fps of RA rate (seconds)

Pilot performed an aggressive maneuver (yes/no)

Average difference of pilot rate to RA directed rate

Absolute Average difference of pilot rate to RA directed rate

Max difference of pilot rate to RA directed rate

Max Rate of Pilot

Total altitude changed over duration of RA (feet)

Absolute Total altitude changed over duration of RA (feet)

Percentage Compliance over duration of the RA with no buffer

Percentage Compliance over duration of the RA with 1/2 fps buffer

Percentage Compliance over duration of the RA with 1 fps buffer

Percentage Compliance over duration of the RA with 2 fps buffer

Percentage Compliance over duration of the RA with 3 fps buffer

RA- Duration Pilot Viewed ND (secs)

RA- Duration Pilot Viewed TSD (secs)

RA- Duration Pilot Viewed PFD (secs)

RA- Duration Pilot Viewed CDU (secs)

RA- Duration Pilot Viewed MCP (secs)

Index of RA in RFS file

Duration of RA (secs)

Last Autopilot Vertical Mode during RA, if changed (numeric)

Last Autopilot Lateral Mode during RA, if changed (numeric)

Autopilot turned on after COC (yes/no)

Time after COC pilot turns back on Autopilot (secs)

Pilot calls ATC after COC (yes/no)

Time after COC pilot calls ATC (secs)

Pilot calls ATC after COC (yes/no), alternate

Time after COC pilot calls ATC (secs)

Index of RFS file for COC

Autopilot Vertical Mode after COC (numeric)

If Verical Mode is Same as Before (yes/no)

Autopilot Vertical Mode Time (secs)

Autopilot Lateral Mode after COC (numeric)

If Lateral Mode is Same as Before (yes/no)

Autopilot Lateral Mode Time (secs)

Return to Clearance

Call ATC for new clearance

APPENDIX E: RANDOM EFFECTS TABLE

Measure	Sig. of Pilot Effects	Pilots Ommitted	Sig. After Omission of Pilots	Sig. of Run Order
Baseline Pilots Compared to Performance During Training				
Time Pilots First Achieved Compliance after RA Initiation	*0.228*			*0.748*
Autopilot Disconnect Time after RA Initiation	0.042	None		*N/A*
Altitude Deviation over Duration of RA	0.013	None		*0.34*
Average Vertical Rate Difference	0.032	301, 316	0.015	*0.28*
Percentage Compliance	*0.896*			*0.781*
Baseline Pilots Compared to Trained Pilots				
Time Pilots First Achieved Compliance after RA Initiation	*0.265*			*0.69*
Autopilot Disconnect Time after RA Initiation	0.01	None		*0.79*
Percentage Compliance	*0.06*			*0.97*
Altitude Deviation over Duration of RA	0.002	301, 316	0.015	*0.82*
Average Vertical Rate Difference	0.002	301, 316	0.028	*0.78*
Maximum Vertical Rate Difference	0.008	316, 318	0.013	*0.55*
Maximum Vertical Rate	0.019	316	0.031	*0.36*
Pilot Response after Clear of Conflict	*N/A*			*N/A*
Pilot Interaction with ATC before the TA	*N/A*			*N/A*
Pilot Interaction with ATC after the TA	*N/A*			*N/A*
Pilot Interaction with ATC after the RA	*N/A*			*N/A*

APPENDIX F: MATLAB DATA ANALYSIS SCRIPTS

RunScripts.m

```
 for i = 1:19       %PILOT NUMBER
     for j = 1:9        %RUN NUMBER


           if i == 5
           %Skip pilot 5. Omit data
         elseif i==8 && j==9
         %elseif i ==7 && j==7

         else



         PilotNumber = num2str(i)
         RunNumber = num2str(j)


         foldername   =   ['../Pilot',PilotNumber,'/Formatted
Data Pilot ', PilotNumber,'/Run',RunNumber];
         cd (foldername);
         filename                                          =
['P',PilotNumber,'R',RunNumber,'Script.txt'];
         fid = fopen(filename);
         A= fscanf(fid,'%s');
         cd('../../../Analysis')
         eval(A)

         end

     end
 end
```

data_analysis.m

```
%%***********************************************************
*********%%
```

```
%%******** 2012 Integrated Simulator, HITL TCAS Study
********%%
%%***************** Data Analysis Package
*****************%%
%%*******************************************************
***%%


% Version 1.0 by Elizabeth Fleming (efleming@gatech.edu)
and Kylie Garrett
% Current Version 4.1 by Elizabeth Fleming
(efleming@gatech.edu)
% Last update Dec 13, 2012

%% IF YOU UPDATE ANY PORTION OF CODE
% RECORD UPDATES IN
TCAS_DataAnalysisScript_Updates.txt

%% Package Scripts:
% data_IO.m : Pulls in relevent data excel files
% Events_time_reorg.m : looks for TCAS advisories
% eventnumber.m : enumerates TCAS advisories
% audio.m : sets time stamp on audio by matching TCAS data
from RFS
% pre_TA.m : summarizes data 30 seconds prior to TA
% TA.m : summarizes data during TA
% RA.m : summarizes data during RA
% clearofconflict : summarized data for 60 seconds after
COC
% data_analysis_TAOnly : run when no RA was incurred for
specific scenario
% RunScripts : outter wrapper to automatically run all
pilots and runs
% TCAS_DataAnalysisScript_Updates.txt : list of script
updates (starting from V4.0)


%% Begin
function data_analysis(PilotNumberRun, RFS_filename,
TCAS_filename, AP_filename, Events_filename,
ET_time_filename, ET_data_filename,
audio_filename,Stat_start)


tic     %Use tic toc to find length of time of process

PilotNumber = num2str(Stat_start(1));
```

```
RunNumber = num2str(Stat_start(2));

foldername = ['../Pilot',PilotNumber,'/Formatted Data Pilot
', PilotNumber,'/Run',RunNumber];
%%calls columns of RFS data from the RFS files
COLUMN_OF_RFS_VSPEED=21;
COLUMN_OF_RFS_ROLL=29;
COLUMN_OF_RFS_COMMANDEDALT=44;
COLUMN_OF_RFS_COMMANDEDHDG=45;
COLUMN_OF_RFS_COMMANDEDAIRSPEED=46;
COLUMN_OF_RFS_COMMANDEDVERTICALSPEED=47;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Data IO--> pulls in the excel files containing data and
converts
% it to a usable Matlab form
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

[RFS_num, RFS_raw, ET_num, ET_raw, TCAS_num, TCAS_raw,
AP_raw, Events_raw, audio_raw]=data_IO(RFS_filename,
TCAS_filename, AP_filename, Events_filename,
ET_time_filename, ET_data_filename, audio_filename,
foldername);


%%
%%%
% Events Reorg
%Adds .1 secs to the events so matlab can distiguish a
change in AP mode%%%%%%%

[Events_reorg]= Events_time_reorg(Events_raw);

%%
%
% Directory Creation
%Creates the directory for data to be saved
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%


dir_name=['./Analysis_',
date,'/DataAnalysis',PilotNumberRun];
```

```
mkdir(dir_name);


%%
%%%%%%%%%Counts the number of TCAS "events"
[TA_start, RA_start, COC_start] = eventnumber(TCAS_num);

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Audio--> Sorts the audio and scans for TCAS events using
the first event
% as a time placeholder
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
TA_1_index = TA_start(1,1);
time_TA_1 = TCAS_num(TA_1_index,1);
[audio_scaled] = audio(time_TA_1, audio_raw);



%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% TA--> Analyzes data for pilot response before a TA and
during TA events
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

run=1;
[TA_rows, TA_cols]= size(TA_start);
for i=1:TA_rows
    TA_ok = TA_start(i, 1);
    filename=['pre_TA_', num2str(i)];
    [PreTA_Statistics] = pre_TA(RFS_num, RFS_raw, ET_num,
ET_raw, TCAS_num, TCAS_raw, AP_raw, Events_reorg, TA_ok,
audio_scaled, filename,dir_name);

    filename=['TA_', num2str(i)];
    [TA_Statistics] = TA(RFS_num, RFS_raw, ET_num, ET_raw,
TCAS_num, TCAS_raw, AP_raw, Events_reorg, TA_ok, RA_start,
audio_scaled, filename,dir_name,PreTA_Statistics);

    CombTAStat(i,:)= [PreTA_Statistics, TA_Statistics];


end
```

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% RA--> Analyzes data for pilot response during RA events
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

run=1;
[RA_rows, RA_cols]= size(RA_start);
for i=1:RA_rows
    RA_ok = RA_start(i, 1);
    filename=['RA_', num2str(i)];
    COC_ok =COC_start(i,:);
    [RA_Stat_Singular]  =  RA(RFS_num,  RFS_raw,  ET_num,
ET_raw,  TCAS_num,  TCAS_raw,  AP_raw,  Events_reorg,  RA_ok,
COC_ok, audio_scaled, filename,dir_name);

    RA_Statistics(i,:) = RA_Stat_Singular  ;


end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Clear of Conflict--> Analyzes data for pilot response
after COC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

run=1;
[COC_rows, COC_cols] = size(COC_start);
for i=1:COC_rows
    COC_ok = COC_start(i,1);
    filename=['COC_', num2str(i)];
    [COC_Stat_Singular]= clearofconflict(RFS_num, RFS_raw,
ET_num, ET_raw, TCAS_num, TCAS_raw, AP_raw,  Events_reorg,
COC_ok, audio_scaled, filename,dir_name);

    COC_Statistics(i,:) = COC_Stat_Singular ;

end

%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Statistics--> Combines all of the output statistics into
one file then
% writes it to a txt file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
run = true;
i = 1;
j = 1;
k=1;

while run == true

    TA_Match = TA_start(i,2);
    RA_Match = RA_start(j,2);
        if TA_Match == RA_Match
            Statistics      (i,:)      =      [Stat_start,i,
CombTAStat(i,:), RA_Statistics(j,:), COC_Statistics(j,:)];
            i = i+1;
            if RA_rows>1
            j=j+1;
            end
        else
            Statistics(i,:)        =        [Stat_start,i,
CombTAStat(i,:), -1, -1,-1, -1,-1, -1,-1, -1,-1, -1,-1,-1,
-1, -1,  -1, -1, -1, -1,-1, -1, -1, -1,-1,-1,-1,-1,-1, -1,-
1, -1, -1, -1,-1, -1, -1, -1, -1, -1, -1, -1, -1,-1,-1, -1,
-1,-1,-1, -1, -1, -1,-1, -1, -1, -1, -1, -1, -1,-1, -1,-1
];
            i = i+1;
        end


        if i> TA_rows
            run = false;
        end
         if j >RA_rows
             run = false;
         end
end


 [Stat_rows, Stat_cols] = size(Statistics);


filename_data=[dir_name,'/Statistics.xls'];
```

```
xlswrite(filename_data, Statistics);
storage = ['./Analysis_', date] ;
cd(storage)
fid=fopen('STATISTICS.txt', 'a');
for i = 1:Stat_rows
fprintf(fid,'%s,', PilotNumberRun);
for j = 1: Stat_cols
fprintf(fid, '%g,',Statistics(i,j)) ;
end
fprintf(fid,'\n');
end

fclose(fid);
cd('../')

toc
end
```

data_IO.m

```
function [RFS_num, RFS_raw, ET_num, ET_raw, TCAS_num,
TCAS_raw, AP_raw, Events_raw,
audio_raw]=data_IO(RFS_filename, TCAS_filename,
AP_filename, Events_filename, ET_time_filename,
ET_data_filename, audio_filename, foldername);
% This function takes in the names of the RFS, ET, and TCAS
data files and
% brings them into the Matlab envoronment. Files are
formatted and output
% as both numerical matricies and cell arrays.

% All _filename inputs are strings in the form 'file.xls'
cd(foldername)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% RFS Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
[RFS_num, RFS_txt, RFS_raw]=xlsread(RFS_filename);
end_time=RFS_num(end,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Eyetracker Data
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
[ET_time_num,                                    ET_time_txt,
ET_time_raw]=xlsread(ET_time_filename);
[ET_data_num,                                    ET_data_txt,
ET_data_raw]=xlsread(ET_data_filename);

% Combining ET time and data files
ET_num=[ET_time_num,ET_data_num];
ET_raw=[ET_time_raw,ET_data_raw];

% ET time column is corrected
[num_points, dum]=size(ET_num);
ET_time_corr_double=[linspace(0, end_time, num_points)]';
ET_time_corr_cellarray=num2cell(ET_time_corr_double);

ET_num(:,2)=ET_time_corr_double;
ET_raw(2:end,2)=ET_time_corr_cellarray;

% filename_data=['./DataAnalysis/eyetracker'];
% xlswrite(filename_data, ET_raw);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% TCAS Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
[TCAS_num, TCAS_txt, TCAS_raw]=xlsread(TCAS_filename);
% It might not make sense to output the _num data because
the file may
% contain valuable non-numerical data. I'm not quite sure
what William
% ment.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% AP Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
[AP_num, AP_txt, AP_raw]=xlsread(AP_filename);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Events Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
```

```
[Events_num,                                           Events_txt,
Events_raw]=xlsread(Events_filename);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Audio Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
[audio_num, audio_txt, audio_raw]=xlsread(audio_filename);

cd('../../../Analysis')
end
```

eventnumber.m

```
function     [TA_start,     RA_start,     COC_start]     =
eventnumber(TCAS_num);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%% TA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% The following code runs multiple TA events
[TCAS_rows, TCAS_cols]=size(TCAS_num);
 TA_start=[];

if TCAS_num(1,2)~=0;
    TA_start=1;
end

% 'TA_start' is the variable that holds the indicies of the
begining of all RA events
for i=2:TCAS_rows
    if TCAS_num(i,2)~=0 && TCAS_num(i-1,2)==0;
        TA_start=[TA_start ; i];
    end
```

```
end

[TArows, TAcols] = size(TA_start);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%% RA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% 'RA_start' is the variable that holds the indicies of the
begining of all RA events
RA_start=[];
if TCAS_num(1,3)~=0;
    RA_start=1;
end

for i=2:TCAS_rows
    if TCAS_num(i,3)~=0 && TCAS_num(i-1,3)==0;

        RA_start=[RA_start ; i];
    end


end

[RArows, RAcols] = size(RA_start);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%% COC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% 'COC' is the variable that holds the indicies of the
begining of all COC events
COC_start =[];

if TCAS_num(1,3)== 1;
    COC_start=1;
end

for i=2:TCAS_rows
    if TCAS_num(i-1,3)~=1 && TCAS_num(i,3)==1;

        COC_start=[COC_start; i];
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%% Correlate TA's and RA's
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
TA_start(:,2) = 0;
RA_start(:,2) = 0;
COC_start(:,2) = 0;

j = 1;
for i = 1:TArows
    TA_start(i,2) = j ;
    j = 1+j;
end


for i = 1:RArows
    RA_place = RA_start(i);
    RA_Start_Time = TCAS_num(RA_place,1);
    for j=1:TArows
        TA_place = TA_start(j);
        TA_Start_Time = TCAS_num(TA_place,1);
        if RA_Start_Time > TA_Start_Time
            RA_start(i,2) = TA_start(j,2);
            COC_start(i,2) = TA_start(j,2);
        end
    end
end
```

Events_time_reorg.m

```
function [Events_reorg]= Events_time_reorg(Events_raw)

h = -1;
Events_time                                                 =
[Events_raw(1,1),Events_raw(1,2),Events_raw(1,3);Events_raw
(2,1),Events_raw(2,2),Events_raw(2,3)];
[rows,cols] = size(Events_raw);

for i = 3:rows

    if Events_raw{i,1} == Events_raw{i-1,1}
        if h == Events_raw{i,1}
```

```
            q = q+0.1;
            Events_new_time= Events_raw{i,1}+q;
            h = Events_raw{i,1};
        else
            q = 0.1;
            Events_new_time= Events_raw{i,1}+q;
           h= Events_raw{i,1};

        end
        Events_time    =    [Events_time;    Events_new_time,
Events_raw(i,2), Events_raw(i,3)];
    else
        Events_time             =             [Events_time;
Events_raw(i,1),Events_raw(i,2),Events_raw(i,3)];
    end
end



Events_reorg = Events_time;

End
```

Audio.m

```
function [audio_scaled] = audio(time_TA_1, audio_raw);

[audio_rows, audio_columns] = size(audio_raw);
run = true;
i = 1;
while run
    check = strcmp('TA', audio_raw{i,3});
    if check ==1
        TA_index=i;
        run = false;
    end
i = i+1;
end
audio_s={};
for i = 1:audio_rows
    audio_scaledtime            =            audio_raw{i,1}-
audio_raw{TA_index,1}+time_TA_1;
    new_row = {audio_scaledtime, audio_raw{i,3}};
    audio_s = [audio_s; new_row];
end
```

```
audio_scaled = audio_s;

end
```

pre_TA.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% Pre-TA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% This function takes in the outputs produced by data_IO and runs data
% analysis on 30 seconds before the TA. Produces an excel file and some
% statistics.

function [PreTA_Statistics] = pre_TA(RFS_num, RFS_raw, ET_num, ET_raw, TCAS_num, TCAS_raw, AP_raw, Events_raw, TA_start, audio_scaled, filename,dir_name)

%Column of relevent data from RFS output
COLUMN_OF_RFS_VSPEED=21;
COLUMN_OF_RFS_ROLL=29;
COLUMN_OF_RFS_COMMANDEDALT=44;
COLUMN_OF_RFS_COMMANDEDHDG=45;
COLUMN_OF_RFS_COMMANDEDAIRSPEED=46;
COLUMN_OF_RFS_COMMANDEDVERTICALSPEED=47;
COLUMN_OF_RFS_ALT = 4;


TA_TellATC={'TA_TellATC'};
epsilon=5;
negepsilon=-5;
run=true;

time_TA=TCAS_raw{TA_start,1};
Vert_Att={};
Pilot_Response_Horiz={};
ClimbRate={};
TurnRate = {};
Altitude = {};
t={};
outputs=[];
```

```
i=1;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%Pre-                    TA
Information%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

% This portion of the code reads through RFS data and
interprets it for
% response information

    while run
    if RFS_num(i,1)>= time_TA-30 & RFS_num(i,1)<= time_TA
        if RFS_num(i,COLUMN_OF_RFS_VSPEED)>5
            Vert_Att='Climb';
        elseif RFS_num(i,COLUMN_OF_RFS_VSPEED)<-5
            Vert_Att='Descend';
        else
            Vert_Att='Level';
        end
        if abs(RFS_num(i,COLUMN_OF_RFS_ROLL)) <= 2
            Pilot_Response_Horiz='No Turn';
        else
            Pilot_Response_Horiz='Turn';
        end
        t=RFS_num(i,1);
        ClimbRate = RFS_num(i,COLUMN_OF_RFS_VSPEED);
        TurnRate = RFS_num(i,COLUMN_OF_RFS_ROLL);
        Altitude = RFS_num(i, COLUMN_OF_RFS_ALT);
        AP_OnOff = AP_raw(i,5);

    elseif RFS_num(i,1) >= time_TA
        run=false;
    end



    if RFS_num(i,1)>= time_TA-30
    new_row = {t, ClimbRate, Vert_Att, Altitude, TurnRate,
Pilot_Response_Horiz, [],[], AP_OnOff, [], []};
    outputs = [outputs; new_row];
    end
    i=i+1;
```

```
end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%Eyetracker%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%Reads through eyetracker data and adds to the RFS data.
Since the time
%stamps do not match exactly between RFS and the
eyetracker, the eyetracker
%data is interpolated for the RFS (ie, if the eyetracker
records something
%at 121.12 and RFS records something at 121,24 then the
eyetracker will
%record their 121.12 finding into the 121.24 RFS data to
keep the data
%consistent.

new_row={};
run=true;
i=2;
time_TA=TCAS_raw{TA_start,1}; %Time TA occurs

%Search through the eyetracker data and capture data that
was recorded
%during the Pre-TA
while run
    if  ET_raw{i,2}>= time_TA-30 & ET_raw{i,2}<= time_TA
        new_row={ET_raw{i,2},[],[],        [],[],        [],
ET_raw{i,8},[], [], [], []};
        outputs=[outputs; new_row];

    elseif ET_raw{i,2} >= time_TA
        run=false;
    end
    i=i+1;
end
```

```
%Combines the RFS and eyetracker data into one file and sorts it
%ascending. Once the data is combined and sorted, the eyetracker data is
%interpolated onto the RFS data.
[sorted, index]=sort([outputs{:,1}]);
outputs=outputs(index,:);
[datarows, datacolumns] = size(outputs);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs(:,7));

while run

    if emptyIndex(i,1)== 1 %checks for empty cells, indicating that RFS did not record durring that time.
        p = i-1;
        outputs(i,7) = outputs(p,7); %replaces the eyetracker data into the blank RFS cell
    else
        outputs(i,7)=outputs(i,7);
    end
    i=i+1;

    if i > datarows
        run = false;
    else
        run = true;
    end

end


run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new(j,:) = outputs(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
```

```
        run = true;
    end

end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%Autopilot%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%Since autopilot data is not continuous, the code has to
recognize
% that the last known mode would have been set prior to RA.
If the mode is
%NOT changed within the advisory, then the mode is the same
as before the
%advisory started. Must hand input last known mode in the
excel file.

%FOR SECOND EVENTS, MUST RESET AP TO THAT OF LAST KNOWN
MODE IF IT IS NOT
%RESET DURING PRE-TA!!!!

new_row={};
run=true;
i=1;
time_TA=TCAS_raw{TA_start,1};

Mode = 'NAV';
Mode_num= 10;
Mode_vert = 'VNAV';
Mode_num_vert= 30;
Mode_lat = 'LNAV';
Mode_num_lat= 29;

[Events_rows, Events_cols] = size(Events_raw);

while run
           i=i+1;
    if i > Events_rows
         run=false;
    elseif Events_raw{i,1} >= time_TA
        run=false;
```

```
    elseif  Events_raw{i,1}<= time_TA
          if Events_raw{i,2} == 26
               Mode = 'FLCH';
            Mode_vert = Mode;
            Mode_num_vert = 26;
          elseif Events_raw{i,2} == 27
               Mode = 'AltHold';
            Mode_vert = Mode;
            Mode_num_vert = 27;
          elseif Events_raw{i,2} == 28
               Mode = 'VS';
            Mode_vert = Mode;
            Mode_num_vert = 28;
          elseif Events_raw{i,2} == 29
               Mode = 'LNAV';
            Mode_lat = Mode;
            Mode_num_lat = 29;
          elseif Events_raw{i,2} == 30
               Mode = 'VNAV';
            Mode_vert = Mode;
            Mode_num_vert = 30;
          elseif Events_raw{i,2} == 31
               Mode = 'Spd';
            Mode_lat = Mode;
            Mode_num_lat = 31;
          elseif Events_raw{i,2} == 32
               Mode = 'HdgHld';
            Mode_lat = Mode;
            Mode_num_lat = 32;
          elseif Events_raw{i,2} == 33
               Mode = 'HdgSlt';
            Mode_lat = Mode;
            Mode_num_lat = 33;
        else
            Mode = 'Null';
          end

        new_row={Events_raw{i,1},[],[],  [],[],  [],  [],[],
[],Events_raw{i,2}, Mode};
        outputs_new=[outputs_new; new_row];


    end


end
```

```
% The following code sorts the RFS and Mode Change data and
outputs
% the data file, similar to how the eyetracker sorts the
data.
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,10));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,10) = outputs_new(j,10);
        outputs_new(j-1,11) = outputs_new(j,11);
    end
end



run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end
```

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Audio%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% The following code sorts the RFS and AP data and outputs
the data file,
% similar to how the eyetracker sorts the data.
new_row={};
run=true;
i=1;
time_TA=TCAS_raw{TA_start,1};
[audio_rows, audio_cols] = size(audio_scaled);
while run
    if          audio_scaled{i,1}>=      time_TA-30      &
audio_scaled{i,1}<= time_TA
        new_row={audio_scaled{i,1},[],[],  [],[],  [],  [],
audio_scaled{i,2},[], [], []};
        outputs_new=[outputs_new; new_row];

    elseif  audio_scaled{i,1}  >=  time_TA   %ET_raw{i,2}  >
time_RA
        run=false;
    end
    i=i+1;
    if i> audio_rows
        run = false;
    end
end



% The following code sorts the RFS and audio data and
outputs the data file
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,8));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,8) = outputs_new(j,8);
```

```
    end
end



run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end



%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%Saving                           to
Excel%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

outputs_new2=['time','PilotClimbRate','PilotVerticalRespons
e',
'Altitude','PilotTurnRate','PilotHorizontalResponse',ET_raw
(1,8),  'Audio','AutoPilot',  'NumericMCPMode',  'MCPMode';
outputs_new2];
filename_data=['./',dir_name,'/',filename, '_Outputs.xls'];
xlswrite(filename_data, outputs_new2);


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%Summarizing                    into
Statistics%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

Pre_TAOutputs = outputs_new2 ;

[rows, columns] = size(Pre_TAOutputs);

PreTA_APOff = 0;
PreTA_CallATC = 0;
run = true;
i = 2;

while run
   AP_Value = Pre_TAOutputs{i,9};

    if  AP_Value{1}== 0;
        PreTA_APOff = 1;
        PreTA_APOff_time      =      Pre_TAOutputs{rows,1}-
Pre_TAOutputs{i,1};
        run = false ;
    else
        PreTA_APOff = 0;
    end

    i = i+1;
    if i>rows
        run = false;
    end
end

run = true;
i = 2;
while run
     check = strcmp('PTTS', Pre_TAOutputs{i,8});

    if check == 1;
        PreTA_CallATC = 1;
        PreTA_CallATC_time          =Pre_TAOutputs{rows,1}-
Pre_TAOutputs{i,1};
        run = false;
    else
        if PreTA_CallATC ~= 1
        PreTA_CallATC = 0;
        end
```

```
    end
    i = i+1;
    if i>rows
        run = false;
    end

end


if PreTA_CallATC == 0;
    PreTA_CallATC_time =-2;
end

if PreTA_APOff == 0;
    PreTA_APOff_time = -2;
end


     check_ND     =     [Pre_TAOutputs{2,1},     strcmp('ND',
Pre_TAOutputs{2,7})];
     check_TSD    =    [Pre_TAOutputs{2,1},    strcmp('TSD',
Pre_TAOutputs{2,7})];
     check_PFD    =    [Pre_TAOutputs{2,1},    strcmp('PFD',
Pre_TAOutputs{2,7})];
     check_CDU    =    [Pre_TAOutputs{2,1},    strcmp('CDU',
Pre_TAOutputs{2,7})];
     check_MCP    =    [Pre_TAOutputs{2,1},    strcmp('MCP',
Pre_TAOutputs{2,7})];


     check_ND_starttime=[];
     check_TSD_starttime=[];
     check_PFD_starttime=[];
     check_CDU_starttime=[];
     check_MCP_starttime=[];
     check_ND_stoptime = [];
     check_TSD_stoptime = [];
     check_PFD_stoptime = [];
     check_MCP_stoptime = [];
     check_CDU_stoptime = [];
     if check_ND(1,2) == 1
         check_ND_starttime = check_ND(1,1);
         size_ND = 1;
     end

     if check_TSD(1,2) == 1
         check_TSD_starttime = check_TSD(1,1);
```

```
     end

     if check_PFD(1,2) == 1
         check_PFD_starttime = check_PFD(1,1);
     end

     if check_CDU(1,2) == 1
         check_CDU_starttime = check_CDU(1,1);
     end

     if check_MCP(1,2) == 1
         check_MCP_starttime = check_MCP(1,1);
     end

for i = 3:rows


     check_ND = [check_ND; Pre_TAOutputs{i,1}, strcmp('ND',
Pre_TAOutputs{i,7})];
     check_TSD     =     [check_TSD;     Pre_TAOutputs{i,1},
strcmp('TSD', Pre_TAOutputs{i,7})];
     check_PFD     =     [check_PFD;     Pre_TAOutputs{i,1},
strcmp('PFD', Pre_TAOutputs{i,7})];
     check_CDU     =     [check_CDU;     Pre_TAOutputs{i,1},
strcmp('CDU', Pre_TAOutputs{i,7})];
     check_MCP     =     [check_MCP;     Pre_TAOutputs{i,1},
strcmp('MCP', Pre_TAOutputs{i,7})];

end

for i = 2:rows-2

    if check_ND(i,2)-check_ND(i-1,2) >0
         check_ND_starttime      =     [check_ND_starttime;
check_ND(i,1)];
    elseif check_ND(i,2)-check_ND(i-1,2) <0
         check_ND_stoptime       =      [check_ND_stoptime;
check_ND(i,1)];
    end

    if check_TSD(i,2)-check_TSD(i-1,2) >0
         check_TSD_starttime     =    [check_TSD_starttime;
check_TSD(i,1)];
    elseif check_TSD(i,2)-check_TSD(i-1,2) <0
         check_TSD_stoptime      =     [check_TSD_stoptime;
check_TSD(i,1)];
    end
```

```
    if check_PFD(i,2)-check_PFD(i-1,2) >0
         check_PFD_starttime     =    [check_PFD_starttime;
check_PFD(i,1)];
    elseif check_PFD(i,2)-check_PFD(i-1,2) <0
         check_PFD_stoptime      =     [check_PFD_stoptime;
check_PFD(i,1)];
    end

    if check_CDU(i,2)-check_CDU(i-1,2) >0
         check_CDU_starttime     =    [check_CDU_starttime;
check_CDU(i,1)];
    elseif check_CDU(i,2)-check_CDU(i-1,2) <0
         check_CDU_stoptime      =     [check_CDU_stoptime;
check_CDU(i,1)];
    end

    if check_MCP(i,2)-check_MCP(i-1,2) >0
         check_MCP_starttime     =    [check_MCP_starttime;
check_MCP(i,1)];
    elseif check_MCP(i,2)-check_MCP(i-1,2) <0
         check_MCP_stoptime      =     [check_MCP_stoptime;
check_MCP(i,1)];
    end
end

[check_ND_starttime_rows,    check_ND_starttime_columns]    =
size(check_ND_starttime);
[check_ND_stoptime_rows,     check_ND_stoptime_columns]     =
size(check_ND_stoptime);
[check_TSD_starttime_rows,   check_TSD_starttime_columns]   =
size(check_TSD_starttime);
[check_TSD_stoptime_rows,    check_TSD_stoptime_columns]    =
size(check_TSD_stoptime);
[check_PFD_starttime_rows,   check_PFD_starttime_columns]   =
size(check_PFD_starttime);
[check_PFD_stoptime_rows,    check_PFD_stoptime_columns]    =
size(check_PFD_stoptime);
[check_CDU_starttime_rows,   check_CDU_starttime_columns]   =
size(check_CDU_starttime);
[check_CDU_stoptime_rows,    check_CDU_stoptime_columns]    =
size(check_CDU_stoptime);
[check_MCP_starttime_rows,   check_MCP_starttime_columns]   =
size(check_MCP_starttime);
[check_MCP_stoptime_rows,    check_MCP_stoptime_columns]    =
size(check_MCP_stoptime);
```

```
if check_ND_starttime_rows>check_ND_stoptime_rows
    check_ND_stoptime         =        [check_ND_stoptime;
check_ND(end,1)];
end

if check_TSD_starttime_rows>check_TSD_stoptime_rows
    check_TSD_stoptime        =       [check_TSD_stoptime;
check_TSD(end,1)];
end

if check_PFD_starttime_rows>check_PFD_stoptime_rows
    check_PFD_stoptime        =       [check_PFD_stoptime;
check_PFD(end,1)];
end

if check_CDU_starttime_rows>check_CDU_stoptime_rows
    check_CDU_stoptime        =       [check_CDU_stoptime;
check_CDU(end,1)];
end


if check_MCP_starttime_rows>check_MCP_stoptime_rows
    check_MCP_stoptime        =       [check_MCP_stoptime;
check_MCP(end,1)];
end


duration_ND_temp=[];
duration_TSD_temp=[];
duration_PFD_temp=[];
duration_CDU_temp=[];
duration_MCP_temp=[];



for i = 1:check_ND_starttime_rows
    duration_ND_temp          =         [duration_ND_temp;
check_ND_stoptime(i,1)-check_ND_starttime(i,1)];
end

for i = 1:check_TSD_starttime_rows
    duration_TSD_temp         =        [duration_TSD_temp;
check_TSD_stoptime(i,1)-check_TSD_starttime(i,1)];
end
```

```
for i = 1:check_PFD_starttime_rows
    duration_PFD_temp          =          [duration_PFD_temp;
check_PFD_stoptime(i,1)-check_PFD_starttime(i,1)];
end

for i = 1:check_CDU_starttime_rows
    duration_CDU_temp          =          [duration_CDU_temp;
check_CDU_stoptime(i,1)-check_CDU_starttime(i,1)];
end


for i = 1:check_MCP_starttime_rows
    duration_MCP_temp          =          [duration_MCP_temp;
check_MCP_stoptime(i,1)-check_MCP_starttime(i,1)];
end

TOTAL_ND_VIEWING = sum(duration_ND_temp);
TOTAL_TSD_VIEWING = sum(duration_TSD_temp);
TOTAL_PFD_VIEWING = sum(duration_PFD_temp);
TOTAL_CDU_VIEWING = sum(duration_CDU_temp);
TOTAL_MCP_VIEWING = sum(duration_MCP_temp);


if TOTAL_ND_VIEWING == 0
    ND_View_YesNo = 0;
else

    ND_View_YesNo = 1;
end


if TOTAL_TSD_VIEWING == 0
    TSD_View_YesNo = 0;
else

    TSD_View_YesNo = 1;
end

if TOTAL_PFD_VIEWING == 0
    PFD_View_YesNo = 0;
else

    PFD_View_YesNo = 1;
end

if TOTAL_CDU_VIEWING == 0
    CDU_View_YesNo = 0;
```

```
else

    CDU_View_YesNo = 1;
end

if TOTAL_MCP_VIEWING == 0
    MCP_View_YesNo = 0;
else

    MCP_View_YesNo = 1;
end




PreTA_Statistics = [PreTA_APOff,PreTA_APOff_time, PreTA_CallATC,PreTA_CallATC_time,
TOTAL_ND_VIEWING,TOTAL_TSD_VIEWING, TOTAL_PFD_VIEWING, TOTAL_CDU_VIEWING, TOTAL_MCP_VIEWING,ND_View_YesNo, TSD_View_YesNo, PFD_View_YesNo, CDU_View_YesNo, MCP_View_YesNo, Mode_num_vert,Mode_num_lat];


End
```

TA.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function takes in the outputs produced by data_IO and runs data
% analysis on TCAS TA. Outputs a text file and some statistics


function [TA_Statistics] = TA(RFS_num, RFS_raw, ET_num, ET_raw, TCAS_num, TCAS_raw, AP_raw, Events_raw, TA_start, RA_start, audio_scaled, filename,dir_name,PreTA_Statistics)

%Column of relevent data from RFS output
COLUMN_OF_RFS_VSPEED=21;
COLUMN_OF_RFS_ROLL=29;
COLUMN_OF_RFS_COMMANDEDALT=44;
```

```
COLUMN_OF_RFS_COMMANDEDHDG=45;
COLUMN_OF_RFS_COMMANDEDAIRSPEED=46;
COLUMN_OF_RFS_COMMANDEDVERTICALSPEED=47;
COLUMN_OF_RFS_ALT = 4;




TA_TellATC={'TA_TellATC'};
epsilon=5;
negepsilon=-5;
run=true;
time_TA=TCAS_raw{TA_start, 1};
Vert_Att={};
Pilot_Response_Horiz={};
ClimbRate={};
TurnRate = {};
t={};
outputs=[];

 [TCAS_rows, TCAS_cols] = size(TCAS_num);

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%TA
Information%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%

% This portion of the code reads through RFS data and
interprets it for
% response information
i=TA_start;
time_TA = RFS_num(i,1);
while run
     if RFS_num(i,1)>= time_TA
        i=i+1;
        if RFS_num(i,COLUMN_OF_RFS_VSPEED)>5
            Vert_Att='Climb';
        elseif RFS_num(i,COLUMN_OF_RFS_VSPEED)<-5
            Vert_Att='Descend';
        else
            Vert_Att='Level';
        end
        if abs(RFS_num(i,COLUMN_OF_RFS_ROLL)) <= 2
```

```
            Pilot_Response_Horiz='No Turn';
        else
            Pilot_Response_Horiz='Turn';
        end
        t=RFS_num(i,1);
        ClimbRate = RFS_num(i,COLUMN_OF_RFS_VSPEED);
        TurnRate = RFS_num(i,COLUMN_OF_RFS_ROLL);
        Altitude = RFS_num(i, COLUMN_OF_RFS_ALT);
        AP_OnOff = AP_raw (i,5);


    if  TCAS_raw{i,3}>=1

        run = false;
        TA_end = i;

    elseif TCAS_raw{i+1,2}< 1
        run = false;
        TA_end = i;

    elseif i >= TCAS_rows
        run=false;
        TA_end = i;
    end

    new_row = {t, ClimbRate, Vert_Att, Altitude, TurnRate,
Pilot_Response_Horiz, [], [], AP_OnOff, [],[]};
    outputs = [outputs; new_row];
    TA_end = i;
    end
end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%Eyetracker%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%Reads through eyetracker data and adds to the RFS data.
Since the time
%stamps do not match exactly between RFS and the
eyetracker, the eyetracker
%data is interpolated for the RFS (ie, if the eyetracker
records something
```

```
%at 121.12 and RFS records something at 121,24 then the
eyetracker will
%record their 121.12 finding into the 121.24 RFS data to
keep the data
%consistent.

new_row={};
run=true;
i=2;
time_TA=TCAS_raw{TA_start,1};%Time TA occurs
time_TAEnd = TCAS_raw{TA_end, 1};%Time TA ends
[ET_rows, ET_cols] = size(ET_raw);

%Search through the eyetracker data and capture data that
was recorded
%during the Pre-TA
while run
    if ET_raw{i,2}>= time_TA & ET_raw{i,2}<= time_TAEnd% &
ET_raw{i,2} <= time_RA)
        new_row={ET_raw{i,2},[],[], [],[], [], ET_raw{i,8},
[], [],[],[]};
        outputs=[outputs; new_row];

    elseif ET_raw{i,2} >= time_TAEnd    %ET_raw{i,2} >
time_RA
        run=false;
    end
    i=i+1;
    if i >= ET_rows
        run = false;
    end

end




%Combines the RFS and eyetracker data into one file and
sorts it
%ascending. Once the data is combined and sorted, the
eyetracker data is
%interpolated onto the RFS data.

[sorted, index]=sort([outputs{:,1}]);
outputs=outputs(index,:);
[datarows, datacolumns] = size(outputs);
```

```
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs(:,7));

while run

    if emptyIndex(i,1)== 1
        p = i-1;
        outputs(i,7) = outputs(p,7);
    else
        outputs(i,7)=outputs(i,7);
    end
    i=i+1;

    if i > datarows
        run = false;
    else
        run = true;
    end


end


run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new(j,:) = outputs(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end



%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Autopilot%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Since autopilot data is not continuous, the code has to recognize
% that the last known mode would have been set prior to RA. If the mode is
%NOT changed within the advisory, then the mode is the same as before the
%advisory started. Must hand input last known mode in the excel file.




new_row={};
run=true;
i=1;
time_TA=TCAS_raw{TA_start,1};
Mode_num=PreTA_Statistics(1,10);
Mode='preTA';
[Events_rows, Events_cols] = size(Events_raw);
Mode_vert = 'preTA';
Mode_num_vert= 30;
Mode_lat = 'preTA';
Mode_num_lat= 29;
while run
           i=i+1;
    if i > Events_rows
            run=false;

    elseif Events_raw{i,1} >= time_TAEnd
        run=false;


%    elseif  Events_raw{i,1}>= time_TA & Events_raw{i,1}<= time_TAEnd
    elseif  Events_raw{i,1}<= time_TAEnd
          if Events_raw{i,2} == 26
              Mode = 'FLCH';
            Mode_vert = Mode;
            Mode_num_vert = 26;
          elseif Events_raw{i,2} == 27
```

```
                Mode = 'AltHold';
             Mode_vert = Mode;
             Mode_num_vert = 27;
           elseif Events_raw{i,2} == 28
                Mode = 'VS';
             Mode_vert = Mode;
             Mode_num_vert = 28;
           elseif Events_raw{i,2} == 29
                Mode = 'LNAV';
             Mode_lat = Mode;
             Mode_num_lat = 29;
           elseif Events_raw{i,2} == 30
                Mode = 'VNAV';
             Mode_vert = Mode;
             Mode_num_vert = 30;
           elseif Events_raw{i,2} == 31
                Mode = 'Spd';
             Mode_lat = Mode;
             Mode_num_lat = 31;
           elseif Events_raw{i,2} == 32
                Mode = 'HdgHld';
             Mode_lat = Mode;
             Mode_num_lat = 32;
           elseif Events_raw{i,2} == 33
                Mode = 'HdgSlt';
             Mode_lat = Mode;
             Mode_num_lat = 33;
         else
             Mode = 'Null';
           end
         new_row={Events_raw{i,1},[],[],  [],[],  [],  [],[],
[],Events_raw{i,2}, Mode};
         outputs_new=[outputs_new; new_row];




    end

end




% The following code sorts the RFS and Mode Change data and
outputs
```

```
% the data file, similar to how the eyetracker sorts the data.


[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,10));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,10) = outputs_new(j,10);
        outputs_new(j-1,11) = outputs_new(j,11);
    end
end



run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end




%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Audio%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% The following code sorts the RFS and AP data and outputs
the data file,
% similar to how the eyetracker sorts the data.


new_row={};
run=true;
i=1;
time_TA=TCAS_raw{TA_start,1};
[audio_rows, audio_cols] = size(audio_scaled);
while run
    if   audio_scaled{i,1}>=  time_TA  &  audio_scaled{i,1}<=
time_TAEnd
        new_row={audio_scaled{i,1},[],[],  [],[],  [],  [],
audio_scaled{i,2}, [],[],[]};
        outputs_new=[outputs_new; new_row];

    elseif audio_scaled{i,1} >=  time_TAEnd
        run=false;
    end
    i=i+1;

    if i > audio_rows
        run = false;
    end
end



% The following code sorts the RFS and audio data and
outputs the data file
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,8));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,8) = outputs_new(j,8);
    end
```

```
end



run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%Saving                            to
Excel%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%


outputs_new2=['time','PilotClimbRate','PilotVerticalRespons
e',
'Altitude','PilotTurnRate','PilotHorizontalResponse',ET_raw
(1,8),  'Audio','AutoPilot',  'NumericMCPMode',  'MCPMode';
outputs_new2];
filename_data=['./',dir_name,'/',filename, '_Outputs.xls'];
xlswrite(filename_data, outputs_new2);

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%Summarizing                         into
Statistics%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%

TAOutputs = outputs_new2 ;

[rows, columns] = size(TAOutputs);
TA_APOff = 0;
TA_CallATC =0;
run = true;

i = 2;

while run
    TA_AP = TAOutputs{i,9};

    if TA_AP{1} == 0;
        TA_APOff = 1;
        TA_APOff_time_afterTA       =       TAOutputs{i,1}-
TAOutputs{2,1};
        TA_APOff_time_beforeRA     =     TAOutputs{rows,1}-
TAOutputs{i,1};
        run = false;
    else
        TA_APOff = 0;

    end

    i = i+1;
    if i>rows
        run = false;
    end

end

i = 2;
run = true;
while run
    check = strcmp('PTTS', TAOutputs{i,8});

    if check == 1;
        TA_CallATC = 1;
        TA_CallATC_time_afterTA      =      TAOutputs{i,1}-
TAOutputs{2,1};
        TA_CallATC_time_beforeRA    =    TAOutputs{rows,1}-
TAOutputs{i,1};
        run = false;
    else
```

```
        if TA_CallATC ~= 1
        TA_CallATC = 0;
        end
    end
    i = i+1;
    if i>rows
        run = false;
    end
end

if TA_CallATC == 0;
    TA_CallATC_time_afterTA = -2;
    TA_CallATC_time_beforeRA = -2;
end

if TA_APOff == 0;
        TA_APOff_time_afterTA = -2;
        TA_APOff_time_beforeRA = -2;
end



     check_ND      =      [TAOutputs{2,1},      strcmp('ND',
TAOutputs{2,7})];
     check_TSD     =     [TAOutputs{2,1},     strcmp('TSD',
TAOutputs{2,7})];
     check_PFD     =     [TAOutputs{2,1},     strcmp('PFD',
TAOutputs{2,7})];
     check_CDU     =     [TAOutputs{2,1},     strcmp('CDU',
TAOutputs{2,7})];
     check_MCP     =     [TAOutputs{2,1},     strcmp('MCP',
TAOutputs{2,7})];


     check_ND_starttime=[];
     check_TSD_starttime=[];
     check_PFD_starttime=[];
     check_CDU_starttime=[];
     check_MCP_starttime=[];
     check_ND_stoptime = [];
     check_TSD_stoptime = [];
     check_PFD_stoptime = [];
     check_MCP_stoptime = [];
     check_CDU_stoptime = [];
     if check_ND(1,2) == 1
         check_ND_starttime = check_ND(1,1);
```

```
        size_ND = 1;
    end

    if check_TSD(1,2) == 1
        check_TSD_starttime = check_TSD(1,1);
    end

    if check_PFD(1,2) == 1
        check_PFD_starttime = check_PFD(1,1);
    end

    if check_CDU(1,2) == 1
        check_CDU_starttime = check_CDU(1,1);
    end

    if check_MCP(1,2) == 1
        check_MCP_starttime = check_MCP(1,1);
    end

for i = 3:rows
%%%%ND

    check_ND  =  [check_ND;  TAOutputs{i,1},  strcmp('ND',
TAOutputs{i,7})];
    check_TSD = [check_TSD; TAOutputs{i,1}, strcmp('TSD',
TAOutputs{i,7})];
    check_PFD = [check_PFD; TAOutputs{i,1}, strcmp('PFD',
TAOutputs{i,7})];
    check_CDU = [check_CDU; TAOutputs{i,1}, strcmp('CDU',
TAOutputs{i,7})];
    check_MCP = [check_MCP; TAOutputs{i,1}, strcmp('MCP',
TAOutputs{i,7})];

end

for i = 2:rows-2

    if check_ND(i,2)-check_ND(i-1,2) >0
        check_ND_starttime      =      [check_ND_starttime;
check_ND(i,1)];
    elseif check_ND(i,2)-check_ND(i-1,2) <0
        check_ND_stoptime       =       [check_ND_stoptime;
check_ND(i,1)];
    end

    if check_TSD(i,2)-check_TSD(i-1,2) >0
```

```
        check_TSD_starttime     =     [check_TSD_starttime;
check_TSD(i,1)];
    elseif check_TSD(i,2)-check_TSD(i-1,2) <0
        check_TSD_stoptime      =      [check_TSD_stoptime;
check_TSD(i,1)];
    end

    if check_PFD(i,2)-check_PFD(i-1,2) >0
        check_PFD_starttime     =     [check_PFD_starttime;
check_PFD(i,1)];
    elseif check_PFD(i,2)-check_PFD(i-1,2) <0
        check_PFD_stoptime      =      [check_PFD_stoptime;
check_PFD(i,1)];
    end

    if check_CDU(i,2)-check_CDU(i-1,2) >0
        check_CDU_starttime     =     [check_CDU_starttime;
check_CDU(i,1)];
    elseif check_CDU(i,2)-check_CDU(i-1,2) <0
        check_CDU_stoptime      =      [check_CDU_stoptime;
check_CDU(i,1)];
    end

    if check_MCP(i,2)-check_MCP(i-1,2) >0
        check_MCP_starttime     =     [check_MCP_starttime;
check_MCP(i,1)];
    elseif check_MCP(i,2)-check_MCP(i-1,2) <0
        check_MCP_stoptime      =      [check_MCP_stoptime;
check_MCP(i,1)];
    end
end

[check_ND_starttime_rows,   check_ND_starttime_columns]   =
size(check_ND_starttime);
[check_ND_stoptime_rows,    check_ND_stoptime_columns]    =
size(check_ND_stoptime);
[check_TSD_starttime_rows,  check_TSD_starttime_columns]  =
size(check_TSD_starttime);
[check_TSD_stoptime_rows,   check_TSD_stoptime_columns]   =
size(check_TSD_stoptime);
[check_PFD_starttime_rows,  check_PFD_starttime_columns]  =
size(check_PFD_starttime);
[check_PFD_stoptime_rows,   check_PFD_stoptime_columns]   =
size(check_PFD_stoptime);
[check_CDU_starttime_rows,  check_CDU_starttime_columns]  =
size(check_CDU_starttime);
```

```
[check_CDU_stoptime_rows,   check_CDU_stoptime_columns]   =
size(check_CDU_stoptime);
[check_MCP_starttime_rows,  check_MCP_starttime_columns]  =
size(check_MCP_starttime);
[check_MCP_stoptime_rows,   check_MCP_stoptime_columns]   =
size(check_MCP_stoptime);



if check_ND_starttime_rows>check_ND_stoptime_rows
    check_ND_stoptime         =         [check_ND_stoptime;
check_ND(end,1)];
end

if check_TSD_starttime_rows>check_TSD_stoptime_rows
    check_TSD_stoptime        =        [check_TSD_stoptime;
check_TSD(end,1)];
end

if check_PFD_starttime_rows>check_PFD_stoptime_rows
    check_PFD_stoptime        =        [check_PFD_stoptime;
check_PFD(end,1)];
end

if check_CDU_starttime_rows>check_CDU_stoptime_rows
    check_CDU_stoptime        =        [check_CDU_stoptime;
check_CDU(end,1)];
end


if check_MCP_starttime_rows>check_MCP_stoptime_rows
    check_MCP_stoptime        =        [check_MCP_stoptime;
check_MCP(end,1)];
end


duration_ND_temp=[];
duration_TSD_temp=[];
duration_PFD_temp=[];
duration_CDU_temp=[];
duration_MCP_temp=[];



for i = 1:check_ND_starttime_rows
    duration_ND_temp          =          [duration_ND_temp;
check_ND_stoptime(i,1)-check_ND_starttime(i,1)];
```

```
end

for i = 1:check_TSD_starttime_rows
    duration_TSD_temp          =          [duration_TSD_temp;
check_TSD_stoptime(i,1)-check_TSD_starttime(i,1)];
end

for i = 1:check_PFD_starttime_rows
    duration_PFD_temp          =          [duration_PFD_temp;
check_PFD_stoptime(i,1)-check_PFD_starttime(i,1)];
end

for i = 1:check_CDU_starttime_rows
    duration_CDU_temp          =          [duration_CDU_temp;
check_CDU_stoptime(i,1)-check_CDU_starttime(i,1)];
end


for i = 1:check_MCP_starttime_rows
    duration_MCP_temp          =          [duration_MCP_temp;
check_MCP_stoptime(i,1)-check_MCP_starttime(i,1)];
end

TOTAL_ND_VIEWING = sum(duration_ND_temp);
TOTAL_TSD_VIEWING = sum(duration_TSD_temp);
TOTAL_PFD_VIEWING = sum(duration_PFD_temp);
TOTAL_CDU_VIEWING = sum(duration_CDU_temp);
TOTAL_MCP_VIEWING = sum(duration_MCP_temp);


if TOTAL_ND_VIEWING == 0
    ND_View_YesNo = 0;
else

    ND_View_YesNo = 1;
end


if TOTAL_TSD_VIEWING == 0
    TSD_View_YesNo = 0;
else

    TSD_View_YesNo = 1;
end

if TOTAL_PFD_VIEWING == 0
    PFD_View_YesNo = 0;
```

```
else

    PFD_View_YesNo = 1;
end

if TOTAL_CDU_VIEWING == 0
    CDU_View_YesNo = 0;
else

    CDU_View_YesNo = 1;
end

if TOTAL_MCP_VIEWING == 0
    MCP_View_YesNo = 0;
else

    MCP_View_YesNo = 1;
end


duration_TA = TAOutputs{rows,1} - TAOutputs{2,1};
pilot_vert_speed_preRA = TAOutputs{rows,2};

TA_Statistics = [TA_APOff, TA_APOff_time_afterTA,
TA_APOff_time_beforeRA, TA_CallATC,
TA_CallATC_time_afterTA, TA_CallATC_time_beforeRA,
TOTAL_ND_VIEWING,TOTAL_TSD_VIEWING, TOTAL_PFD_VIEWING,
TOTAL_CDU_VIEWING, TOTAL_MCP_VIEWING,ND_View_YesNo,
TSD_View_YesNo, PFD_View_YesNo, CDU_View_YesNo,
MCP_View_YesNo, TA_start, duration_TA,
pilot_vert_speed_preRA, Mode_num_vert, Mode_num_lat];

end
```

RA.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% RA --> Resolution Advisory
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% This function takes in the outputs produced by data_IO
and runs data
% analysis on TCAS RA Alerts. Outputs a text file and some
statistics
```

```
function [RA_Statistics] = RA(RFS_num, RFS_raw, ET_num,
ET_raw, TCAS_num, TCAS_raw, AP_raw, Events_raw, RA_start,
COC, audio_scaled, filename, dir_name);


[TCAS_rows, TCAS_cols]=size(TCAS_num);
[RFS_rows, RFS_cols]=size(RFS_num);
[ET_rows, ET_cols]=size(ET_raw);

%Column of relevent data from RFS output
COLUMN_OF_RFS_VSPEED=21;
COLUMN_OF_RFS_ROLL=29;
COLUMN_OF_RFS_COMMANDEDALT=44;
COLUMN_OF_RFS_COMMANDEDHDG=45;
COLUMN_OF_RFS_COMMANDEDAIRSPEED=46;
COLUMN_OF_RFS_COMMANDEDVERTICALSPEED=47;
COLUMN_OF_RFS_ALT = 4;
COLUMN_OF_RFS_HEADING = 31;

time_RA = TCAS_raw{RA_start,1};
time_COC = TCAS_raw{COC, 1};
run = true;
time = time_RA;

alerttype = {};
TCAS_vertspeed = {};
ClimbRate = {};
TurnRate = {};
Altitude = {};
Vert_Att={};
Pilot_Response_Horiz={};
time = {};
outputs = {};
alert_output= {};
epsilon=5;
negepsilon=-5;
TCAS_pos=0;

alerttype_1 = TCAS_num(RA_start,3);


Holdtime = TCAS_num(RA_start,1)+5;
```

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%RA
Information%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%This portion of the code reads through RFS data and
interprets it for
%RA type, vertical rate of advisory and pilot, compliance,
%aggressiveness, etc

for i = RA_start:COC;

    time = TCAS_raw{i,1};

    if RFS_num(i,COLUMN_OF_RFS_VSPEED)>5
            Vert_Att='Climb';
        elseif RFS_num(i,COLUMN_OF_RFS_VSPEED)<-5
            Vert_Att='Descend';
        else %if RFS_num(i,COLUMN_OF_RFS_VSPEED)==0
            Vert_Att='Level';
    end
    if abs(RFS_num(i,COLUMN_OF_RFS_ROLL)) <= 5
            Pilot_Response_Horiz='No Turn';
            Pilot_H = 0;
    else
            Pilot_Response_Horiz='Turn';
            if time>Holdtime;
                Pilot_H = 1;
            else
                Pilot_H = 0;
            end

    end

        ClimbRate = RFS_num(i,COLUMN_OF_RFS_VSPEED);
        TurnRate = RFS_num(i,COLUMN_OF_RFS_ROLL);
        Altitude = RFS_num(i, COLUMN_OF_RFS_ALT);
        Heading = RFS_num(i, COLUMN_OF_RFS_HEADING);
        AP_OnOff = AP_raw {i,5};

        TCAS_vertspeed = TCAS_num(i,8);
```

```
    %Using  the  RFS_RA  file,  feed  in  the  advisory  code
(Columns 3-7). The vertical speed is in col 8.
    alerttype_1 = TCAS_num(i,3);    %RA Type, Active
    alerttype_2 = TCAS_num(i,4);     %RA Vertical (Crossing,
reversal, strengthen, weaken)
    alerttype_3 = TCAS_num(i,5);    %Crossing binary
    alerttype_4 = TCAS_num(i,6);    %do not descend
    alerttype_5 = TCAS_num(i,7);    %do not climb

    %Each col represents a definition of the advisory type.
Read through
    %each    column    and    assign    it's    corresponding
interpretation

    %Possible  RA  types.  Initially  set  all  values  to  0.
Values will be reset
    %depending on RFS RA data

    active_climb = 0 ;
    active_descend = 0;
    preventive = 0;
    crossing= 0;
    reversal = 0;
    increase = 0;
    maintain = 0;
    RA_type = '';
    alert_output_numeric = 0;

    %Col  3  (alerttype_1)  is  the  Combined  Control  (Active
Climb/Descend or
    %Preventive)
    if alerttype_1 == 4
        RA_combined_control = 'Corrective, Climb';
        active_climb = 1 ;
        RA_type = 'Climb' ;
        alert_output_numeric = 1;

    elseif alerttype_1 == 5
        RA_combined_control = 'Corrective, Descend';
        active_descend = 1;
        RA_type = 'Descend' ;
        alert_output_numeric = 2;

     elseif alerttype_1 == 6
        RA_combined_control = 'Preventive';
        preventive = 1;
        RA_type = 'Preventive' ;
```

```
        alert_output_numeric = 3;
    elseif alerttype_1 == 1

    else
        error
    end

    %Col  4  (alerttype_2)  assigns  the  specific  RA  type
(Crossing, Reversal,
    %Strengthening, Maintain)

    if alerttype_2 == 0
        RA_Vertical = 'NA' ;

    elseif alerttype_2 == 1
        RA_type = strcat('Crossing_', RA_type);
        alert_output_numeric = alert_output_numeric+3;
        %Crossing Climb = 4, Crossing Descend = 5, Crossing
Preventive = 6

    elseif  alerttype_2 == 2
        RA_type = strcat('Reversal_', RA_type);
        alert_output_numeric = alert_output_numeric+6;
        %Reversal Climb = 7, Reversal descend = 8, Reversal
Preventive = 9

    elseif  alerttype_2 == 3
        RA_type = strcat('Increase_', RA_type);
        alert_output_numeric = alert_output_numeric+9;
        %Increase  Climb  =  10,  Increase  descend  =  11,
Increase Preventive = 12

    elseif alerttype_2 == 4
        RA_type = strcat('Maintain_', RA_type);
        alert_output_numeric = alert_output_numeric+9;
        %Maintain  Climb  =  13,  Maintain  descend  =  14,
Maintain Preventive = 15

    else
        error

    end


        %Col  5  (alerttype_3)  assigns  crossing  or  not
(binary 0-1)
    if alerttype_3==1
```

```
        if alerttype_2 == 0
        RA_type = strcat('Crossing_', RA_type);
        alert_output_numeric = alert_output_numeric+3;
        %Crossing Climb = 4, Crossing Descend = 5, Crossing
Preventive = 6
        end
    end

    %Initially set values to 0
    preventive_up = 0;
    preventive_down = 0;

     %Col 6 (alerttype_4) assigns up advisory specifics

    %Set preventive RA features (do not descend, do not
descend > ...)
            %Preventive_up = 1, Do not descend
            %Preventive_up = 2, Do not descend Greater than
500 fpm
            %Preventive_up = 3, Do Not Descend Greater than
1000 fpm
            %Preventive_up = 2, Do not descend Greater than
1500 fpm

    if alerttype_4 == 0
        RA_up = 'NA (Down Advisory)';
    elseif alerttype_4 == 1
        RA_up = 'Climb';
    elseif alerttype_4 == 2
        RA_up = 'Do Not Descend';
        if alerttype_1 == 6
            preventive_up = 1;
        else
            preventive_up = 0;
        end
    elseif alerttype_4 == 3
        RA_up = 'Do Not Descend Greater than 500 fpm';
        if alerttype_1 == 6
            preventive_up = 2;
        else
            preventive_up = 0;
        end
    elseif alerttype_4 == 4
        RA_up = 'Do Not Descend Greater than 1000 fpm';
        if alerttype_1 == 6
            preventive_up = 3;
        else
```

```
            preventive_up = 0;
        end
    elseif alerttype_4 == 5
        RA_up = 'Do Not Descend Greater than 2000 fpm';
        if alerttype_1 == 6
            preventive_up = 4;
        else
            preventive_up = 0;
        end
    else
        error    %breakout if this is not correct
    end

    %Col 7 (alerttype_5) assigns down advisory specifics

    %Set  preventive  RA  features  (do  not  climb   ,  do  not
climb > ...)
            %Preventive_down = 1, Do not climb
            %Preventive_down = 2, Do not climb Greater than
500 fpm
            %Preventive_down = 3, Do Not climb Greater than
1000 fpm
            %Preventive_down = 2, Do not climb Greater than
1500 fpm
    if alerttype_5 == 0
        RA_up = 'NA (Up Advisory)';
    elseif alerttype_5 == 1
        RA_up = 'Descend';
    elseif alerttype_5 == 2
        RA_up = 'Do Not Climb';
        if alerttype_1 == 6
            preventive_down = 1;
        else
            preventive_down = 0;
        end
    elseif alerttype_5 == 3
        RA_up = 'Do Not Climb Greater than 500 fpm';
        if alerttype_1 == 6
            preventive_down = 2;
        else
            preventive_down = 0;
        end
    elseif alerttype_5 == 4
        RA_up = 'Do Not Climb Greater than 1000 fpm';
        if alerttype_1 == 6
            preventive_down = 3;
        else
```

```
            preventive_down = 0;
        end
    elseif alerttype_5 == 5
        RA_up = 'Do Not Climb Greater than 2000 fpm';
        if alerttype_1 == 6
            preventive_down = 4;
        else
            preventive_down = 0;
        end

    else
        error
    end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%Compliance                    and
Aggressive%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%


    if TCAS_num(i,1)>Holdtime       %5 second hold time for
initial RA, 2.5 sec for subsequent advisories
        if alerttype_1 == 4          %CLIMB RA, pilot vert
speed needs to be greater than TCAS for compliance
            if  ClimbRate  <  TCAS_vertspeed      %if  pilot
vertical  speed  (ClimbRate)  is  LESS  THAN  TCAS_vertspeed,
pilot is not in compliance
                Comp_Rate_0 = 0;         %Comp_Rate_0 = 0 ==
for no buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_0 = 1;        %in Compliance
            end

            if  ClimbRate  <  TCAS_vertspeed  -0.5   %0.5  fps
buffer
                Comp_Rate_5 = 0;         %Comp_Rate_5 = 0 ==
for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_5 = 1;        %in Compliance
            end

            if ClimbRate < TCAS_vertspeed -1 %1 fps buffer
```

```
                Comp_Rate_1 = 0;        %Comp_Rate_1 = 0 ==
for 1 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_1 = 1;        %in Compliance
            end

            if  ClimbRate  <  TCAS_vertspeed  -2      %2  fps
buffer
                Comp_Rate_2 = 0;        %Comp_Rate_2 = 0 ==
for 2 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_2 = 1;        %in Compliance
            end

            if ClimbRate < TCAS_vertspeed -3  %3 fps buffer
                Comp_Rate_3 = 0;        %Comp_Rate_3 = 0 ==
for 3 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_3 = 1;        %in Compliance
            end

            %Set  Aggressive,  TRUE  if  climb  rate  is  10  fps
greater than abs
            %value of TCAS vertspeed
            if ClimbRate >TCAS_vertspeed +10
                Comp_A = 1;
            else
                Comp_A = 0;
            end


        elseif  alerttype_1  ==  5        %%Descend  RA,  pilot
vert speed needs to be less than TCAS for compliance
           if  ClimbRate  >  TCAS_vertspeed      %if  pilot
vertical  speed  (ClimbRate)  is  GREATER  THAN  TCAS_vertspeed,
pilot is not in compliance
                Comp_Rate_0 = 0;        %Comp_Rate_0 = 0 ==
for no buffer, the pilot is NOT IN COMPLIANCE
           else
                Comp_Rate_0 = 1;        %in Compliance
           end

           if  ClimbRate  >  TCAS_vertspeed  +  0.5  %0.5  fps
buffer
                Comp_Rate_5 = 0;        %Comp_Rate_5 = 0 ==
for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
           else
```

```
                Comp_Rate_5 = 1;        %in Compliance
            end

            if  ClimbRate  >  TCAS_vertspeed  +  1      %1  fps
buffer
                Comp_Rate_1 = 0;        %Comp_Rate_1 = 0 ==
for 1 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_1 = 1;        %in Compliance
            end

            if  ClimbRate  >  TCAS_vertspeed  +  2      %2  fps
buffer
                Comp_Rate_2 = 0;        %Comp_Rate_2 = 0 ==
for 2 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_2 = 1;        %in Compliance
            end

            if  ClimbRate  >  TCAS_vertspeed  +  3      %3  fps
buffer
                Comp_Rate_3 = 0;        %Comp_Rate_3 = 0 ==
for 3 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_3 = 1;        %in Compliance
            end

            %Set  Aggressive,  TRUE  if  climb  rate  is  10  fps
greater than abs
             %value of TCAS vertspeed
             if ClimbRate <TCAS_vertspeed -10
                Comp_A = 1;
             else
                Comp_A = 0;
             end

        elseif  alerttype_1 == 6
            if alerttype_5 ==0
               %Set rate required to comply
                if alerttype_4== 1
                        Comply_Rate = 0 ;          %Climb
                elseif alerttype_4 == 2
                        Comply_Rate = 0 ;           %Do not
descend
                elseif alerttype_4 == 3
                        Comply_Rate  =  -500/60  ;        %To
comply to the advisory, do not descend greater than 500 fpm
```

```
                elseif alerttype_4 == 4
                        Comply_Rate = -1000/60 ;        %To
comply to the advisory, do not descend greater than 1000
fpm
                elseif alerttype_4 == 5
                        Comply_Rate = -2000/60 ;
                else
                    error                  %To comply to the
advisory, do not descend greater than 2000 fpm
                end
                %Check for compliance
                if ClimbRate < Comply_Rate      %if pilot
vertical speed (ClimbRate) is greater than compliance rate,
pilot is not in compliance
                    Comp_Rate_0 = 0;          %Comp_Rate_0 =
0 == for no buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_0 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -0.5   %0.5 fps
buffer
                    Comp_Rate_5 = 0;          %Comp_Rate_5 =
0 == for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_5 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -1 %1 fps buffer
                    Comp_Rate_1 = 0;          %Comp_Rate_1 =
0 == for 1 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_1 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -2      %2 fps
buffer
                    Comp_Rate_2 = 0;          %Comp_Rate_2 =
0 == for 2 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_2 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -3    %3 fps
buffer
                    Comp_Rate_3 = 0;          %Comp_Rate_3 =
0 == for 3 fps buffer, the pilot is NOT IN COMPLIANCE
```

```
                else
                    Comp_Rate_3 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -10
                    Comp_A = 1;
                else
                    Comp_A = 0;
                end

            elseif alerttype_4 ==0
               %Set rate required to comply
                if alerttype_5 == 1
                        Comply_Rate       =       0       ;
%descend
                elseif alerttype_5 == 2
                        Comply_Rate = 0 ;            %Do not
climb
                elseif alerttype_5 == 3
                        Comply_Rate  =  500/60  ;         %To
comply to the advisory, do not climb greater than 500 fpm
                elseif alerttype_5 == 4
                        Comply_Rate  =  1000/60  ;        %To
comply to the advisory, do not climb greater than 1000 fpm
                elseif alerttype_5 == 5
                        Comply_Rate  =  2000/60  ;        %To
comply to the advisory, do not climb greater than 2000 fpm
                else
                    error
                end
                %Check for compliance
                if  ClimbRate  >  Comply_Rate     %if  pilot
vertical  speed  (ClimbRate)  is  less  than  compliance  rate,
pilot is not in compliance
                    Comp_Rate_0 = 0;          %Comp_Rate_0 =
0 == for no buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_0 = 1;        %in Compliance
                end

                if ClimbRate > Comply_Rate  +0.5  %0.5 fps
buffer
                    Comp_Rate_5 = 0;          %Comp_Rate_5 =
0 == for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_5 = 1;        %in Compliance
                end
```

```
                if  ClimbRate  >  Comply_Rate    +1  %1  fps
buffer
                    Comp_Rate_1 = 0;         %Comp_Rate_1 =
0 == for 1 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_1 = 1;        %in Compliance
                end

                if  ClimbRate  >  Comply_Rate   +2     %2  fps
buffer
                    Comp_Rate_2 = 0;         %Comp_Rate_2 =
0 == for 2 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_2 = 1;        %in Compliance
                end

                if  ClimbRate  >  Comply_Rate    +3    %3  fps
buffer
                    Comp_Rate_3 = 0;         %Comp_Rate_3 =
0 == for 3 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_3 = 1;        %in Compliance
                end

                if ClimbRate > Comply_Rate +10
                    Comp_A = 1;
                else
                    Comp_A = 0;
                end

            elseif alerttype_1 == 1
            Comp_A = 0;
            Comp_Rate_0 = 1;
            Comp_Rate_5 = 1;
            Comp_Rate_1 = 1;
            Comp_Rate_2 = 1;
            Comp_Rate_3 = 1;
            else
                error
            end
        end
    else
            Comp_A = 0;
            Comp_Rate_0 = 1;
            Comp_Rate_5 = 1;
            Comp_Rate_1 = 1;
```

```
            Comp_Rate_2 = 1;
            Comp_Rate_3 = 1;
    end





    %If the RA type changes, add "hold time"
    if TCAS_num(i,8) == TCAS_num(i-1,8)
        Change_RA = 0;
        RA_vert_orig = TCAS_num(i,8);
        RA_vert_new = 0;
    elseif TCAS_num(i,8) ~= TCAS_num(i-1,8)
        Change_RA = 1;
        Holdtime_new = TCAS_num(i,1)+2.5;
        if Holdtime_new>Holdtime;         %If the RA resets
before the original holdtime ends then keep larger hold
time.
            Holdtime = Holdtime_new;
        else
            Holdtime = Holdtime;
        end
            if TCAS_num(i,8) ~= -1
                RA_vert_orig = TCAS_num(i-1,8);
                RA_vert_new = TCAS_num(i,8);
            end
    end

    %record values
    new_row = {time, ClimbRate, Vert_Att, Altitude,
TurnRate, Heading, Pilot_Response_Horiz, [],
TCAS_vertspeed, alert_output, [], Pilot_H , Comp_A,
Comp_Rate_0,Comp_Rate_5,Comp_Rate_1,Comp_Rate_2,Comp_Rate_3
, AP_OnOff,[],[], alert_output_numeric};

    %Compile over time by adding new rows
    outputs = [outputs; new_row];

end



%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%Eyetracker%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%Reads through eyetracker data and adds to the RFS data. Since the time
%stamps do not match exactly between RFS and the eyetracker, the eyetracker
%data is interpolated for the RFS (ie, if the eyetracker records something
%at 121.12 and RFS records something at 121,24 then the eyetracker will
%record their 121.12 finding into the 121.24 RFS data to keep the data
%consistent.

new_row={};
run=true;
i=2;
time_RA=TCAS_raw{RA_start,1};  %Time RA occurs

%Search through the eyetracker data and capture data that was recorded
%during the RA
while run
    if ET_raw{i,2}>= time_RA & ET_raw{i,2}<= time_COC

        new_row = {ET_raw{i,2}, [], [], [], [], [], [], ET_raw{i,8}, [], [], [], [], [], [],[],[],[],[], [],[],[], []};     %feed in eyetracker data
        outputs=[outputs; new_row];

    elseif ET_raw{i,2} > time_COC
        run=false;
    end

    i=i+1;
end



%Combines the RFS and eyetracker data into one file and sorts it
```

```
%ascending.  Once  the  data  is  combined  and  sorted,  the
eyetracker data is
%interpolated onto the RFS data.
[sorted, index]=sort([outputs{:,1}]);
outputs=outputs(index,:);
[datarows, datacolumns] = size(outputs);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs(:,8));

while run

    if  emptyIndex(i,1)==  1    %checks  for  empty  cells,
indicating that RFS did not record durring that time.
        p = i-1;
        outputs(i,8)    =    outputs(p,8);   %replaces   the
eyetracker data into the blank RFS cell
    else
        outputs(i,8)=outputs(i,8);
    end
    i=i+1;

    if i > datarows
        run = false;
    else
        run = true;
    end

end


run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new(j,:) = outputs(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end
```

```
end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%Autopilot%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%Since autopilot data is not continuous, the code has to
recognize
% that the last known mode would have been set prior to RA.
If the mode is
%NOT changed within the advisory, then the mode is the same
as before the
%advisory started. Must hand input last known mode in the
excel file.

new_row={};
run=true;
i=1;
Mode = 'TA';
Mode_num = 10;
[Events_rows, Events_cols] = size(Events_raw);
Mode_vert = 'TA';
Mode_num_vert= 30;
Mode_lat = 'TA';
Mode_num_lat= 29;
while run
    i=i+1;
  if i > Events_rows
         run=false;

   elseif  Events_raw{i,1}  >=  time_COC    %ET_raw{i,2}  >
time_RA
        run=false;


%     elseif   Events_raw{i,1}>= time_RA & Events_raw{i,1}<=
time_COC
    elseif  Events_raw{i,1}<= time_COC
          if Events_raw{i,2} == 26
               Mode = 'FLCH';
            Mode_vert = Mode;
            Mode_num_vert = 26;
```

```
            elseif Events_raw{i,2} == 27
                Mode = 'AltHold';
              Mode_vert = Mode;
              Mode_num_vert = 27;
            elseif Events_raw{i,2} == 28
                Mode = 'VS';
              Mode_vert = Mode;
              Mode_num_vert = 28;
            elseif Events_raw{i,2} == 29
                Mode = 'LNAV';
              Mode_lat = Mode;
              Mode_num_lat = 29;
            elseif Events_raw{i,2} == 30
                Mode = 'VNAV';
              Mode_vert = Mode;
              Mode_num_vert = 30;
            elseif Events_raw{i,2} == 31
                Mode = 'Spd';
              Mode_lat = Mode;
              Mode_num_lat = 31;
            elseif Events_raw{i,2} == 32
                Mode = 'HdgHld';
              Mode_lat = Mode;
              Mode_num_lat = 32;
            elseif Events_raw{i,2} == 33
                Mode = 'HdgSlt';
              Mode_lat = Mode;
              Mode_num_lat = 33;
        else
            Mode = 'Null';
            end
        new_row = {Events_raw{i,1}, [], [], [], [], [], [],
[], [], [], [], [], [], [],[],[],[],[], [],Events_raw{i,2},
Mode, []};

        outputs_new=[outputs_new; new_row];
  end


end

LastMode = Mode;
LastMode_num = Mode_num;

% The following code sorts the RFS and Mode Change data and
outputs
```

```
% the data file, similar to how the eyetracker sorts the
data.
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,20));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,20) = outputs_new(j,20);
        outputs_new(j-1,21) = outputs_new(j,21);
    end
end



run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Audio%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% The following code sorts the RFS and AP data and outputs
the data file,
```

```
% similar to how the eyetracker sorts the data.
new_row={};
run=true;
i=1;
time_RA=TCAS_raw{RA_start,1};
[audio_rows, audio_cols] = size(audio_scaled);

while run
    if   audio_scaled{i,1}>=  time_RA  &  audio_scaled{i,1}<=
time_COC
        new_row = {audio_scaled{i,1}, [], [], [], [], [],
[], audio_scaled{i,2}, [], [], [], [], [], [],[],[],[],[],
[],[],[], []};
        outputs_new=[outputs_new; new_row];

    elseif audio_scaled{i,1} >= time_COC
        run=false;

    end
    i=i+1;
    if i> audio_rows
        run = false;
    end
end



% The following code sorts the RFS and audio data and
outputs the data file
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,8));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,8) = outputs_new(j,8);
    end
end



run = true;
i=1;
```

```
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;
[datarows, datacolumns] = size(outputs_new);
for i = 1:datarows
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end

end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%Saving                            to
Excel%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

outputs_new2=['time','PilotClimbRate','PilotVerticalRespons
e',       'Altitude','PilotTurnRate',       'PilotHeading',
'PilotHorizontalResponse',ET_raw(1,8),     'TCAS_vertspeed',
'RA_Type',    'Audio',    'Horizontal',    'Agressiveness',
'Compliance_Rate_0','Compliance_Rate_5',
'Compliance_Rate_1','Compliance_Rate_2','Compliance_Rate_3'
,'AutoPilot','NumericMode','Mode','RATypeNumeric';
outputs_new2];
RA_outputs = outputs_new2;

%%Record the data to a text file. The excel function is
really crappy for
%%this code, for an unknown reason. Must use fprintf
instead of xlswrite.

filename_data=['./',dir_name,'/',filename, '_Outputs.txt'];

fid=fopen(filename_data,'w');
for i = 1:length(outputs_new2(:,1))
    for j = 1:length(outputs_new2(1,:))
        k = class(outputs_new2{i,j});
        if isempty(outputs_new2{i,j})
            fprintf(fid, 'empty,');
        elseif strcmp(k,'double')
            fprintf(fid, '%f,',outputs_new2{i,j});
        elseif strcmp(k,'str')
            fprintf(fid, '%f,',outputs_new2{i,j});
```

```
            elseif strcmp(k,'char')
                fprintf(fid, '%s,',outputs_new2{i,j});
            elseif strcmp(k,'cell')
                fprintf(fid, '%s,',outputs_new2{i,j});
            else
                fprintf(fid, '%s,',outputs_new2{i,j});
            end
        end
            fprintf(fid, '\n');
end
fclose(fid);




%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%Summarizing                        into
Statistics%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%% Checking for compliance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%

time_RA=TCAS_raw{RA_start,1};   %start at time of RA
RA_DURATION = time_COC-time_RA ;    %RA duration

[rows, columns] = size(RA_outputs);

%initially set everything equal to 0
RA_APOff = 0;
RA_CallATC = 0;
RA_Horiz_Man =0;
RA_Vert_Comp = 0;
RA_Agressive = 0;
RA_diff = [];
pilot_rate = [];
RA_Rate_0 = 0;
RA_Rate_1 = 0;
RA_Rate_2 = 0;
RA_Rate_5 = 0;
```

```
RA_Rate_3 = 0;

%%
% RA_outputs = {time, ClimbRate, Vert_Att, Altitude,
TurnRate, Heading, Pilot_Response_Horiz, [],
TCAS_vertspeed, alert_output, [], Pilot_H , Comp_A,
Comp_Rate_0,Comp_Rate_5,Comp_Rate_1,Comp_Rate_2,Comp_Rate_3
, AP_OnOff,[],[], alert_output_numeric};

%Exactly matched vertical rate NO BUFFER
time_comp_start_0 = {time_RA};
time_comp_stop_0 = {};

time_comp_start_5 = {time_RA};
time_comp_stop_5 = {};

time_comp_start_1 = {time_RA};
time_comp_stop_1 = {};

time_comp_start_2 = {time_RA};
time_comp_stop_2 = {};

time_comp_start_3 = {time_RA};
time_comp_stop_3 = {};


%for the entire file (1-rows), looking for if the pilot
complied.
for i = 3:rows

    pilot_rate = [pilot_rate; RA_outputs{i,2}]; %col 2 ==
pilot vert speed
    difference = RA_outputs{i,2}-RA_outputs{i,9}; %col
9== TCAS_Vert Speed, this takes the difference from the
TCAS vert speed and the pilot's vert speed
    RA_diff = [RA_diff; difference]; %saves the vert
speed difference


%looking for where the bit flips from a 1 to a 0 reason why
we do it
% this way: for the first 5 seconds the pilot is always in
compliance
% due to RA logic (assumes 5 sec response time).Looking for
the bit flip
% tells us how long after, including those 5 secs the pilot
took to comply.
```

```
% If the pilot started in compliance and remained in compliance then we can
% see that as well.
%if the pilot was complying (1) and then stopped complying (0) then
% changes = -1 (<0). If the pilot was not complying (1) and then started
% complying, then changes = 1 (>0)
    changes_0 = RA_outputs{i,14} - RA_outputs{i-1,14} ; %comply current row with no buffer - comply previous row with no buffer
    changes_5 = RA_outputs{i,15} - RA_outputs{i-1,15} ; %comply current row with .5fps buffer - comply previous row with .5fps buffer
    changes_1 = RA_outputs{i,16} - RA_outputs{i-1,16} ; %comply current row with 1fps buffer - comply previous row with 1fps buffer
    changes_2 = RA_outputs{i,17} - RA_outputs{i-1,17} ; %comply current row with 2fps buffer - comply previous row with 2fps buffer
    changes_3 = RA_outputs{i,18} - RA_outputs{i-1,18} ; %comply current row with 3fps buffer - comply previous row with 3fps buffer



    if changes_0 > 0 %looks for when the pilot was complying and then stopped complying
            time_comp_start_0 = [time_comp_start_0; RA_outputs{i,1}];

    elseif changes_0 < 0 %looks for when the pilot was not complying and then started complying
           time_comp_stop_0 = [time_comp_stop_0; RA_outputs{i,1}];
    end


    if changes_5 > 0 %looks for when the pilot was complying and then stopped complying
            time_comp_start_5 = [time_comp_start_5; RA_outputs{i,1}];

    elseif changes_5 < 0 %looks for when the pilot was not complying and then started complying
```

```
           time_comp_stop_5       =       [time_comp_stop_5;
RA_outputs{i,1}];
    end

    if  changes_1  >  0  %looks  for  when  the  pilot  was
complying and then stopped complying
            time_comp_start_1     =     [time_comp_start_1;
RA_outputs{i,1}];

    elseif changes_1 < 0 %looks for when the pilot was not
complying and then started complying
           time_comp_stop_1       =       [time_comp_stop_1;
RA_outputs{i,1}];
    end

     if  changes_2  >  0  %looks  for  when  the  pilot  was
complying and then stopped complying
            time_comp_start_2     =     [time_comp_start_2;
RA_outputs{i,1}];

    elseif changes_2 < 0 %looks for when the pilot was not
complying and then started complying
           time_comp_stop_2       =       [time_comp_stop_2;
RA_outputs{i,1}];
     end

    if  changes_3  >  0  %looks  for  when  the  pilot  was
complying and then stopped complying
            time_comp_start_3     =     [time_comp_start_3;
RA_outputs{i,1}];

     elseif changes_3 < 0 %looks for when the pilot was not
complying and then started complying
           time_comp_stop_3       =       [time_comp_stop_3;
RA_outputs{i,1}];
    end



end

RA_diff_average = mean(RA_diff);    %Average vertical speed
difference
RA_diff_max = max(abs(RA_diff),[],1);   %max vertical speed
difference
pilot_rate_max  =  max(abs(pilot_rate),[],1);          %max
vertical speed
```

```
abs_RA_diff_average = abs(RA_diff_average);


%%%%%%%%%%%%%Percentage    Compliance    with    NO    Buffer
%%%%%%%%%%%%%%%%%%%%%
 [starttimes_0, dummy] = size(time_comp_start_0);
 [stoptimes_0, dummy] = size(time_comp_stop_0);

Compliance_Block_0 = [];

for i=1:(starttimes_0-1)
    Compliance_Block_0        =        [Compliance_Block_0;
time_comp_stop_0{i,1}-time_comp_start_0{i,1}];
end

if starttimes_0 == stoptimes_0
    Compliance_Block_0        =        [Compliance_Block_0;
time_comp_stop_0{end,1}-time_comp_start_0{end,1}];
elseif starttimes_0 > stoptimes_0
   Compliance_Block_0   =   [Compliance_Block_0;   time_COC-
time_comp_start_0{end,1}];
else
    error
end

Compliance_Total_Time_0 = sum(Compliance_Block_0);
percentage_compliance_0                                    =
Compliance_Total_Time_0/RA_DURATION     *100;   %time   pilot
complied/total time of RA == percentage compliance

if starttimes_0 > 1
    RA_MatchRate_0 = time_comp_start_0{2,1}-time_RA;
elseif starttimes_0 == 1
    if stoptimes_0 == 0
        RA_MatchRate_0 = 0 ;
    elseif stoptimes_0 == 1
        RA_MatchRate_0 = -2;
    else
        error
    end
end

if percentage_compliance_0 == 100
    RA_Rate_0 =1;
else
    RA_Rate_0 =0;
```

```
end

%%%%%%%%%%%%%Percentage  Compliance  with  0.5  FPS  Buffer
%%%%%%%%%%%%%%%%%%%%%%
 [starttimes_5, dummy] = size(time_comp_start_5);
 [stoptimes_5, dummy] = size(time_comp_stop_5);

Compliance_Block_5 = [];

for i=1:(starttimes_5-1)
    Compliance_Block_5        =        [Compliance_Block_5;
time_comp_stop_5{i,1}-time_comp_start_5{i,1}];
end

if starttimes_5 == stoptimes_5
    Compliance_Block_5        =        [Compliance_Block_5;
time_comp_stop_5{end,1}-time_comp_start_5{end,1}];
elseif starttimes_5 > stoptimes_5
   Compliance_Block_5   =   [Compliance_Block_5;   time_COC-
time_comp_start_5{end,1}];
else
    error
end

Compliance_Total_Time_5 = sum(Compliance_Block_5);
percentage_compliance_5                                    =
Compliance_Total_Time_5/RA_DURATION     *100;   %time   pilot
complied/total time of RA == percentage compliance

if starttimes_5 > 1
    RA_MatchRate_5 = time_comp_start_5{2,1}-time_RA;
elseif starttimes_5 == 1
    if stoptimes_5 == 0
        RA_MatchRate_5 = 0 ;
    elseif stoptimes_5 == 1
        RA_MatchRate_5 = -2;
    else
        error
    end
end

if percentage_compliance_5 == 100
    RA_Rate_5 =1;
else
    RA_Rate_5 =0;
end
```

```
%%%%%%%%%%%%%Percentage Compliance with 1 FPS Buffer
%%%%%%%%%%%%%%%%%%%%%
 [starttimes_1, dummy] = size(time_comp_start_1);
 [stoptimes_1, dummy] = size(time_comp_stop_1);

Compliance_Block_1 = [];

for i=1:(starttimes_1-1)
    Compliance_Block_1 = [Compliance_Block_1;
time_comp_stop_1{i,1}-time_comp_start_1{i,1}];
end

if starttimes_1 == stoptimes_1
    Compliance_Block_1 = [Compliance_Block_1;
time_comp_stop_1{end,1}-time_comp_start_1{end,1}];
elseif starttimes_1 > stoptimes_1
   Compliance_Block_1 = [Compliance_Block_1; time_COC-
time_comp_start_1{end,1}];
else
    error
end

Compliance_Total_Time_1 = sum(Compliance_Block_1);
percentage_compliance_1 =
Compliance_Total_Time_1/RA_DURATION *100; %time pilot
complied/total time of RA == percentage compliance


if starttimes_1 > 1
    RA_MatchRate_1 = time_comp_start_1{2,1}-time_RA;
elseif starttimes_1 == 1
    if stoptimes_1 == 0
        RA_MatchRate_1 = 0 ;
    elseif stoptimes_1 == 1
        RA_MatchRate_1 = -2;
    else
        error
    end
end

if percentage_compliance_1 == 100
    RA_Rate_1 =1;
else
    RA_Rate_1 =0;
end
%%%%%%%%%%%%%Percentage Compliance with 2 FPS Buffer
%%%%%%%%%%%%%%%%%%%%%
```

```
 [starttimes_2, dummy] = size(time_comp_start_2);
 [stoptimes_2, dummy] = size(time_comp_stop_2);

Compliance_Block_2 = [];

for i=1:(starttimes_2-1)
    Compliance_Block_2        =        [Compliance_Block_2;
time_comp_stop_2{i,1}-time_comp_start_2{i,1}];
end

if starttimes_2 == stoptimes_2
    Compliance_Block_2        =        [Compliance_Block_2;
time_comp_stop_2{end,1}-time_comp_start_2{end,1}];
elseif starttimes_2 > stoptimes_2
   Compliance_Block_2   =   [Compliance_Block_2;   time_COC-
time_comp_start_2{end,1}];
else
    error
end

Compliance_Total_Time_2 = sum(Compliance_Block_2);
percentage_compliance_2                                     =
Compliance_Total_Time_2/RA_DURATION     *100;  %time  pilot
complied/total time of RA == percentage compliance

if starttimes_2 > 1
    RA_MatchRate_2 = time_comp_start_2{2,1}-time_RA;
elseif starttimes_2 == 1
    if stoptimes_2 == 0
        RA_MatchRate_2 = 0 ;
    elseif stoptimes_2 == 1
        RA_MatchRate_2 = -2;
    else
        error
    end
end

if percentage_compliance_2 == 100
    RA_Rate_2 =1;
else
    RA_Rate_2 =0;
end
%%%%%%%%%%%%%Percentage  Compliance  with  3  FPS  Buffer
%%%%%%%%%%%%%%%%%%%%%
 [starttimes_3, dummy] = size(time_comp_start_3);
 [stoptimes_3, dummy] = size(time_comp_stop_3);
```

```
Compliance_Block_3 = [];

for i=1:(starttimes_3-1)
    Compliance_Block_3        =        [Compliance_Block_3;
time_comp_stop_3{i,1}-time_comp_start_3{i,1}];
end

if starttimes_3 == stoptimes_3
    Compliance_Block_3        =        [Compliance_Block_3;
time_comp_stop_3{end,1}-time_comp_start_3{end,1}];
elseif starttimes_3 > stoptimes_3
   Compliance_Block_3   =   [Compliance_Block_3;   time_COC-
time_comp_start_3{end,1}];
else
    error
end

Compliance_Total_Time_3 = sum(Compliance_Block_3);
percentage_compliance_3                                            =
Compliance_Total_Time_3/RA_DURATION     *100;  %time  pilot
complied/total time of RA == percentage compliance

if starttimes_3 > 1
    RA_MatchRate_3 = time_comp_start_3{2,1}-time_RA;
elseif starttimes_3 == 1
    if stoptimes_3 == 0
        RA_MatchRate_3 = 0 ;
    elseif stoptimes_3 == 1
        RA_MatchRate_3 = -2;
    else
        error
    end
end

if percentage_compliance_3 == 100
    RA_Rate_3 =1;
else
    RA_Rate_3 =0;
end

%% Binary values and Others are set below this line

[rows, columns] = size(RA_outputs);
[AP_rows, AP_cols] = size(AP_raw);


alt1 = RA_outputs{2,4};
```

```
alt2 = RA_outputs{rows,4};
AltitudeChange = alt2 - alt1;

hdg1 = RA_outputs{2,6};
hdg2 = RA_outputs{rows,6};
HeadingChange = hdg2-hdg1;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
%%%%%%%%%%%%%           Binary          Horizontal          and
Aggressive%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
if HeadingChange >=15
    RA_Horiz_Man = 1;
end

skip_H = 0;
skip_V = 0;
skip_A = 0;
skip_0 = 0;
skip_5 = 0;
skip_1 = 0;
skip_2 = 0;
skip_3 = 0;



for i = 2:rows;

    RA_Horiz = RA_outputs{i, 12};
    if RA_Horiz == 1 && skip_H == 0;
        RA_Horiz_Man = 1;
        skip_H = 1;
    elseif RA_Horiz_Man ~= 1 && skip_H == 0;
        RA_Horiz_Man = 0;
    end

      RA_Ag = RA_outputs{i, 13};
    if RA_Ag == 1 && skip_A ==0;
        RA_Agressive = 1;
        skip_A= 1;
    elseif RA_Agressive ~= 1 && skip_A ==0;
        RA_Agressive = 0;

    end
```

```
end



%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%AutoPilot%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%did the pilot turn their AP off? And, at what time?
run = true;
i = 2;
while run
    RA_AP =  RA_outputs{i,19};
    if RA_AP == 0;
        RA_APOff = 1;
        RA_APOff_time = RA_outputs{i,1}-RA_outputs{2,1};
        run = false;
    else
        RA_APOff = 0;

    end
    i = i+1;
    if i>rows
        run = false;
    end
end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%Call ATC%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Call ATC? What time?

run = true;
i = 2;
while run

    check = strcmp('PTTS', RA_outputs{i,11});
    if check == 1;
        RA_CallATC = 1;
```

```
        RA_CallATC_time = RA_outputs{i,1}-RA_outputs{2,1};
        run = false;
    else
        if RA_CallATC ~=1
        RA_CallATC = 0;
        end
    end

    i = i+1;
    if i>rows
        run = false;
    end
end

if RA_CallATC == 0;
    RA_CallATC_time = -2;
end

if RA_APOff == 0;
    RA_APOff_time = -2;
end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%Eyetracker%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Did the pilot look at a specific display? For how long
over duration of event?
     check_ND     =     [RA_outputs{2,1},      strcmp('ND',
RA_outputs{2,8})];
     check_TSD     =     [RA_outputs{2,1},     strcmp('TSD',
RA_outputs{2,8})];
     check_PFD     =     [RA_outputs{2,1},     strcmp('PFD',
RA_outputs{2,8})];
     check_CDU     =     [RA_outputs{2,1},     strcmp('CDU',
RA_outputs{2,8})];
     check_MCP     =     [RA_outputs{2,1},     strcmp('MCP',
RA_outputs{2,8})];


     check_ND_starttime=[];
     check_TSD_starttime=[];
     check_PFD_starttime=[];
     check_CDU_starttime=[];
     check_MCP_starttime=[];
```

```
     check_ND_stoptime = [];
     check_TSD_stoptime = [];
     check_PFD_stoptime = [];
     check_MCP_stoptime = [];
     check_CDU_stoptime = [];

     %Record if the pilot looked at the display
     if check_ND(1,2) == 1
         check_ND_starttime = check_ND(1,1);
         size_ND = 1;
     end

     if check_TSD(1,2) == 1
         check_TSD_starttime = check_TSD(1,1);
     end

     if check_PFD(1,2) == 1
         check_PFD_starttime = check_PFD(1,1);
     end

     if check_CDU(1,2) == 1
         check_CDU_starttime = check_CDU(1,1);
     end

     if check_MCP(1,2) == 1
         check_MCP_starttime = check_MCP(1,1);
     end

for i = 3:rows


     check_ND = [check_ND; RA_outputs{i,1}, strcmp('ND',
RA_outputs{i,8})];
     check_TSD = [check_TSD; RA_outputs{i,1}, strcmp('TSD',
RA_outputs{i,8})];
     check_PFD = [check_PFD; RA_outputs{i,1}, strcmp('PFD',
RA_outputs{i,8})];
     check_CDU = [check_CDU; RA_outputs{i,1}, strcmp('CDU',
RA_outputs{i,8})];
     check_MCP = [check_MCP; RA_outputs{i,1}, strcmp('MCP',
RA_outputs{i,8})];

end

for i = 2:rows-2

    if check_ND(i,2)-check_ND(i-1,2) >0
```

```
        check_ND_starttime      =      [check_ND_starttime;
check_ND(i,1)];
    elseif check_ND(i,2)-check_ND(i-1,2) <0
        check_ND_stoptime       =       [check_ND_stoptime;
check_ND(i,1)];
    end

    if check_TSD(i,2)-check_TSD(i-1,2) >0
        check_TSD_starttime     =     [check_TSD_starttime;
check_TSD(i,1)];
    elseif check_TSD(i,2)-check_TSD(i-1,2) <0
        check_TSD_stoptime      =      [check_TSD_stoptime;
check_TSD(i,1)];
    end

    if check_PFD(i,2)-check_PFD(i-1,2) >0
        check_PFD_starttime     =     [check_PFD_starttime;
check_PFD(i,1)];
    elseif check_PFD(i,2)-check_PFD(i-1,2) <0
        check_PFD_stoptime      =      [check_PFD_stoptime;
check_PFD(i,1)];
    end

    if check_CDU(i,2)-check_CDU(i-1,2) >0
        check_CDU_starttime     =     [check_CDU_starttime;
check_CDU(i,1)];
    elseif check_CDU(i,2)-check_CDU(i-1,2) <0
        check_CDU_stoptime      =      [check_CDU_stoptime;
check_CDU(i,1)];
    end

    if check_MCP(i,2)-check_MCP(i-1,2) >0
        check_MCP_starttime     =     [check_MCP_starttime;
check_MCP(i,1)];
    elseif check_MCP(i,2)-check_MCP(i-1,2) <0
        check_MCP_stoptime      =      [check_MCP_stoptime;
check_MCP(i,1)];
    end
end

[check_ND_starttime_rows,   check_ND_starttime_columns]   =
size(check_ND_starttime);
[check_ND_stoptime_rows,    check_ND_stoptime_columns]    =
size(check_ND_stoptime);
[check_TSD_starttime_rows,  check_TSD_starttime_columns]  =
size(check_TSD_starttime);
```

```
[check_TSD_stoptime_rows,   check_TSD_stoptime_columns]   =
size(check_TSD_stoptime);
[check_PFD_starttime_rows,  check_PFD_starttime_columns]  =
size(check_PFD_starttime);
[check_PFD_stoptime_rows,   check_PFD_stoptime_columns]   =
size(check_PFD_stoptime);
[check_CDU_starttime_rows,  check_CDU_starttime_columns]  =
size(check_CDU_starttime);
[check_CDU_stoptime_rows,   check_CDU_stoptime_columns]   =
size(check_CDU_stoptime);
[check_MCP_starttime_rows,  check_MCP_starttime_columns]  =
size(check_MCP_starttime);
[check_MCP_stoptime_rows,   check_MCP_stoptime_columns]   =
size(check_MCP_stoptime);



if check_ND_starttime_rows>check_ND_stoptime_rows
    check_ND_stoptime         =         [check_ND_stoptime;
check_ND(end,1)];
end

if check_TSD_starttime_rows>check_TSD_stoptime_rows
    check_TSD_stoptime        =        [check_TSD_stoptime;
check_TSD(end,1)];
end

if check_PFD_starttime_rows>check_PFD_stoptime_rows
    check_PFD_stoptime        =        [check_PFD_stoptime;
check_PFD(end,1)];
end

if check_CDU_starttime_rows>check_CDU_stoptime_rows
    check_CDU_stoptime        =        [check_CDU_stoptime;
check_CDU(end,1)];
end


if check_MCP_starttime_rows>check_MCP_stoptime_rows
    check_MCP_stoptime        =        [check_MCP_stoptime;
check_MCP(end,1)];
end


duration_ND_temp=[];
duration_TSD_temp=[];
duration_PFD_temp=[];
```

```
duration_CDU_temp=[];
duration_MCP_temp=[];



for i = 1:check_ND_starttime_rows
    duration_ND_temp          =          [duration_ND_temp;
check_ND_stoptime(i,1)-check_ND_starttime(i,1)];
end

for i = 1:check_TSD_starttime_rows
    duration_TSD_temp         =         [duration_TSD_temp;
check_TSD_stoptime(i,1)-check_TSD_starttime(i,1)];
end

for i = 1:check_PFD_starttime_rows
    duration_PFD_temp         =         [duration_PFD_temp;
check_PFD_stoptime(i,1)-check_PFD_starttime(i,1)];
end

for i = 1:check_CDU_starttime_rows
    duration_CDU_temp         =         [duration_CDU_temp;
check_CDU_stoptime(i,1)-check_CDU_starttime(i,1)];
end


for i = 1:check_MCP_starttime_rows
    duration_MCP_temp         =         [duration_MCP_temp;
check_MCP_stoptime(i,1)-check_MCP_starttime(i,1)];
end

TOTAL_ND_VIEWING = sum(duration_ND_temp);
TOTAL_TSD_VIEWING = sum(duration_TSD_temp);
TOTAL_PFD_VIEWING = sum(duration_PFD_temp);
TOTAL_CDU_VIEWING = sum(duration_CDU_temp);
TOTAL_MCP_VIEWING = sum(duration_MCP_temp);

%Binary display values
if TOTAL_ND_VIEWING == 0
    ND_View_YesNo = 0;
else

    ND_View_YesNo = 1;
end


if TOTAL_TSD_VIEWING == 0
```

```
    TSD_View_YesNo = 0;
else

    TSD_View_YesNo = 1;
end

if TOTAL_PFD_VIEWING == 0
    PFD_View_YesNo = 0;
else

    PFD_View_YesNo = 1;
end

if TOTAL_CDU_VIEWING == 0
    CDU_View_YesNo = 0;
else

    CDU_View_YesNo = 1;
end

if TOTAL_MCP_VIEWING == 0
    MCP_View_YesNo = 0;
else

    MCP_View_YesNo = 1;
end



duration_RA = RA_outputs{rows,1}- RA_outputs{2,1};
time_RA=TCAS_raw{RA_start,1};
RA_DURATION = time_COC-time_RA ;


absAltChange = abs(AltitudeChange);

%%
RAType = cell2mat(RA_outputs(2,22));

%%

%OUTPUTS!!%
RA_Statistics     =     [RAType,     RA_APOff,RA_APOff_time,
RA_CallATC,RA_CallATC_time,      RA_Horiz_Man,HeadingChange,
RA_Rate_0,RA_Rate_5,          RA_Rate_1,RA_Rate_2,RA_Rate_3,
RA_MatchRate_0,     RA_MatchRate_5,          RA_MatchRate_1,
RA_MatchRate_2,                              RA_MatchRate_3,
```

```
RA_Agressive,abs_RA_diff_average, RA_diff_max, pilot_rate_max,
absAltChange,percentage_compliance_0,percentage_compliance_5,percentage_compliance_1,percentage_compliance_2,percentage_compliance_3, TOTAL_ND_VIEWING,TOTAL_TSD_VIEWING, TOTAL_PFD_VIEWING, TOTAL_CDU_VIEWING, TOTAL_MCP_VIEWING,ND_View_YesNo, TSD_View_YesNo, PFD_View_YesNo, CDU_View_YesNo, MCP_View_YesNo, RA_start, duration_RA, Mode_num_vert, Mode_num_lat];

 end
```

clearofconflict.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RA --> Resolution Advisory
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function takes in the outputs produced by data_IO and runs data
% analysis on TCAS RA Alerts. Outputs a text file and some statistics



function [RA_Statistics] = RA(RFS_num, RFS_raw, ET_num, ET_raw, TCAS_num, TCAS_raw, AP_raw, Events_raw, RA_start, COC, audio_scaled, filename, dir_name);


[TCAS_rows, TCAS_cols]=size(TCAS_num);
[RFS_rows, RFS_cols]=size(RFS_num);
[ET_rows, ET_cols]=size(ET_raw);

%Column of relevent data from RFS output
COLUMN_OF_RFS_VSPEED=21;
COLUMN_OF_RFS_ROLL=29;
COLUMN_OF_RFS_COMMANDEDALT=44;
COLUMN_OF_RFS_COMMANDEDHDG=45;
COLUMN_OF_RFS_COMMANDEDAIRSPEED=46;
COLUMN_OF_RFS_COMMANDEDVERTICALSPEED=47;
COLUMN_OF_RFS_ALT = 4;
```

```
COLUMN_OF_RFS_HEADING = 31;

time_RA = TCAS_raw{RA_start,1};
time_COC = TCAS_raw{COC, 1};
run = true;
time = time_RA;

alerttype = {};
TCAS_vertspeed = {};
ClimbRate = {};
TurnRate = {};
Altitude = {};
Vert_Att={};
Pilot_Response_Horiz={};
time = {};
outputs = {};
alert_output= {};
epsilon=5;
negepsilon=-5;
TCAS_pos=0;

alerttype_1 = TCAS_num(RA_start,3);


Holdtime = TCAS_num(RA_start,1)+5;


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%RA
Information%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%This portion of the code reads through RFS data and
interprets it for
%RA type, vertical rate of advisory and pilot, compliance,
%aggressiveness, etc

for i = RA_start:COC;

    time = TCAS_raw{i,1};

    if RFS_num(i,COLUMN_OF_RFS_VSPEED)>5
            Vert_Att='Climb';
        elseif RFS_num(i,COLUMN_OF_RFS_VSPEED)<-5
```

```
            Vert_Att='Descend';
        else %if RFS_num(i,COLUMN_OF_RFS_VSPEED)==0
            Vert_Att='Level';
    end
    if abs(RFS_num(i,COLUMN_OF_RFS_ROLL)) <= 5
            Pilot_Response_Horiz='No Turn';
            Pilot_H = 0;
    else
            Pilot_Response_Horiz='Turn';
            if time>Holdtime;
                Pilot_H = 1;
            else
                Pilot_H = 0;
            end

    end

        ClimbRate = RFS_num(i,COLUMN_OF_RFS_VSPEED);
        TurnRate = RFS_num(i,COLUMN_OF_RFS_ROLL);
        Altitude = RFS_num(i, COLUMN_OF_RFS_ALT);
        Heading = RFS_num(i, COLUMN_OF_RFS_HEADING);
        AP_OnOff = AP_raw {i,5};

        TCAS_vertspeed = TCAS_num(i,8);


    %Using the RFS_RA file, feed in the advisory code
(Columns 3-7). The vertical speed is in col 8.
    alerttype_1 = TCAS_num(i,3);    %RA Type, Active
    alerttype_2 = TCAS_num(i,4);     %RA Vertical (Crossing,
reversal, strengthen, weaken)
    alerttype_3 = TCAS_num(i,5);    %Crossing binary
    alerttype_4 = TCAS_num(i,6);    %do not descend
    alerttype_5 = TCAS_num(i,7);    %do not climb

    %Each col represents a definition of the advisory type.
Read through
    %each column and assign it's corresponding
interpretation

    %Possible RA types. Initially set all values to 0.
Values will be reset
    %depending on RFS RA data

    active_climb = 0 ;
    active_descend = 0;
    preventive = 0;
```

```
    crossing= 0;
    reversal = 0;
    increase = 0;
    maintain = 0;
    RA_type = '';
    alert_output_numeric = 0;

    %Col  3  (alerttype_1)  is  the  Combined  Control  (Active
Climb/Descend or
    %Preventive)
    if alerttype_1 == 4
        RA_combined_control = 'Corrective, Climb';
        active_climb = 1 ;
        RA_type = 'Climb' ;
        alert_output_numeric = 1;

    elseif alerttype_1 == 5
        RA_combined_control = 'Corrective, Descend';
        active_descend = 1;
        RA_type = 'Descend' ;
        alert_output_numeric = 2;

     elseif alerttype_1 == 6
        RA_combined_control = 'Preventive';
        preventive = 1;
        RA_type = 'Preventive' ;
        alert_output_numeric = 3;
    elseif alerttype_1 == 1

    else
        error
    end

    %Col  4  (alerttype_2)  assigns  the  specific  RA  type
(Crossing, Reversal,
    %Strengthening, Maintain)

    if alerttype_2 == 0
        RA_Vertical = 'NA' ;

    elseif alerttype_2 == 1
        RA_type = strcat('Crossing_', RA_type);
        alert_output_numeric = alert_output_numeric+3;
        %Crossing Climb = 4, Crossing Descend = 5, Crossing
Preventive = 6

    elseif  alerttype_2 == 2
```

```
        RA_type = strcat('Reversal_', RA_type);
        alert_output_numeric = alert_output_numeric+6;
        %Reversal Climb = 7, Reversal descend = 8, Reversal
Preventive = 9

    elseif  alerttype_2 == 3
        RA_type = strcat('Increase_', RA_type);
        alert_output_numeric = alert_output_numeric+9;
        %Increase  Climb  =  10,  Increase  descend  =  11,
Increase Preventive = 12

    elseif alerttype_2 == 4
        RA_type = strcat('Maintain_', RA_type);
        alert_output_numeric = alert_output_numeric+9;
        %Maintain  Climb  =  13,  Maintain  descend  =  14,
Maintain Preventive = 15

    else
        error

    end


        %Col  5  (alerttype_3)  assigns  crossing  or  not
(binary 0-1)
    if alerttype_3==1
        if alerttype_2 == 0
        RA_type = strcat('Crossing_', RA_type);
        alert_output_numeric = alert_output_numeric+3;
        %Crossing Climb = 4, Crossing Descend = 5, Crossing
Preventive = 6
        end
    end

    %Initially set values to 0
    preventive_up = 0;
    preventive_down = 0;

     %Col 6 (alerttype_4) assigns up advisory specifics

    %Set  preventive  RA  features  (do  not  descend,  do  not
descend > ...)
            %Preventive_up = 1, Do not descend
            %Preventive_up = 2, Do not descend Greater than
500 fpm
            %Preventive_up = 3, Do Not Descend Greater than
1000 fpm
```

```
            %Preventive_up = 2, Do not descend Greater than
1500 fpm

    if alerttype_4 == 0
        RA_up = 'NA (Down Advisory)';
    elseif alerttype_4 == 1
        RA_up = 'Climb';
    elseif alerttype_4 == 2
        RA_up = 'Do Not Descend';
        if alerttype_1 == 6
            preventive_up = 1;
        else
            preventive_up = 0;
        end
    elseif alerttype_4 == 3
        RA_up = 'Do Not Descend Greater than 500 fpm';
        if alerttype_1 == 6
            preventive_up = 2;
        else
            preventive_up = 0;
        end
    elseif alerttype_4 == 4
        RA_up = 'Do Not Descend Greater than 1000 fpm';
        if alerttype_1 == 6
            preventive_up = 3;
        else
            preventive_up = 0;
        end
    elseif alerttype_4 == 5
        RA_up = 'Do Not Descend Greater than 2000 fpm';
        if alerttype_1 == 6
            preventive_up = 4;
        else
            preventive_up = 0;
        end
    else
        error    %breakout if this is not correct
    end

    %Col 7 (alerttype_5) assigns down advisory specifics

    %Set preventive RA features (do not climb   , do not
climb > ...)
            %Preventive_down = 1, Do not climb
            %Preventive_down = 2, Do not climb Greater than
500 fpm
```

```
            %Preventive_down = 3, Do Not climb Greater than
1000 fpm
            %Preventive_down = 2, Do not climb Greater than
1500 fpm
    if alerttype_5 == 0
        RA_up = 'NA (Up Advisory)';
    elseif alerttype_5 == 1
        RA_up = 'Descend';
    elseif alerttype_5 == 2
        RA_up = 'Do Not Climb';
        if alerttype_1 == 6
            preventive_down = 1;
        else
            preventive_down = 0;
        end
    elseif alerttype_5 == 3
        RA_up = 'Do Not Climb Greater than 500 fpm';
        if alerttype_1 == 6
            preventive_down = 2;
        else
            preventive_down = 0;
        end
    elseif alerttype_5 == 4
        RA_up = 'Do Not Climb Greater than 1000 fpm';
        if alerttype_1 == 6
            preventive_down = 3;
        else
            preventive_down = 0;
        end
    elseif alerttype_5 == 5
        RA_up = 'Do Not Climb Greater than 2000 fpm';
        if alerttype_1 == 6
            preventive_down = 4;
        else
            preventive_down = 0;
        end

    else
        error
    end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%Compliance                    and
Aggressive%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

    if TCAS_num(i,1)>Holdtime       %5 second hold time for
initial RA, 2.5 sec for subsequent advisories
        if alerttype_1 == 4          %CLIMB RA, pilot vert
speed needs to be greater than TCAS for compliance
            if ClimbRate < TCAS_vertspeed     %if pilot
vertical speed (ClimbRate) is LESS THAN TCAS_vertspeed,
pilot is not in compliance
                Comp_Rate_0 = 0;        %Comp_Rate_0 = 0 ==
for no buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_0 = 1;        %in Compliance
            end

            if ClimbRate < TCAS_vertspeed -0.5   %0.5 fps
buffer
                Comp_Rate_5 = 0;        %Comp_Rate_5 = 0 ==
for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_5 = 1;        %in Compliance
            end

            if ClimbRate < TCAS_vertspeed -1 %1 fps buffer
                Comp_Rate_1 = 0;        %Comp_Rate_1 = 0 ==
for 1 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_1 = 1;        %in Compliance
            end

            if ClimbRate < TCAS_vertspeed -2      %2 fps
buffer
                Comp_Rate_2 = 0;        %Comp_Rate_2 = 0 ==
for 2 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_2 = 1;        %in Compliance
            end

            if ClimbRate < TCAS_vertspeed -3  %3 fps buffer
                Comp_Rate_3 = 0;        %Comp_Rate_3 = 0 ==
for 3 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_3 = 1;        %in Compliance
            end
```

```
            %Set Aggressive, TRUE if climb rate is 10 fps
greater than abs
            %value of TCAS vertspeed
            if ClimbRate >TCAS_vertspeed +10
                Comp_A = 1;
            else
                Comp_A = 0;
            end


        elseif alerttype_1 == 5        %%Descend RA, pilot
vert speed needs to be less than TCAS for compliance
           if ClimbRate > TCAS_vertspeed       %if pilot
vertical speed (ClimbRate) is GREATER THAN TCAS_vertspeed,
pilot is not in compliance
                Comp_Rate_0 = 0;         %Comp_Rate_0 = 0 ==
for no buffer, the pilot is NOT IN COMPLIANCE
           else
                Comp_Rate_0 = 1;        %in Compliance
           end

           if ClimbRate > TCAS_vertspeed + 0.5 %0.5 fps
buffer
                Comp_Rate_5 = 0;         %Comp_Rate_5 = 0 ==
for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
           else
                Comp_Rate_5 = 1;        %in Compliance
           end

           if ClimbRate > TCAS_vertspeed + 1      %1 fps
buffer
                Comp_Rate_1 = 0;         %Comp_Rate_1 = 0 ==
for 1 fps buffer, the pilot is NOT IN COMPLIANCE
           else
                Comp_Rate_1 = 1;        %in Compliance
           end

           if ClimbRate > TCAS_vertspeed + 2      %2 fps
buffer
                Comp_Rate_2 = 0;         %Comp_Rate_2 = 0 ==
for 2 fps buffer, the pilot is NOT IN COMPLIANCE
           else
                Comp_Rate_2 = 1;        %in Compliance
           end
```

```
            if  ClimbRate  >  TCAS_vertspeed  +  3      %3  fps
buffer
                 Comp_Rate_3 = 0;        %Comp_Rate_3 = 0 ==
for 3 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                 Comp_Rate_3 = 1;        %in Compliance
            end

            %Set  Aggressive,  TRUE  if  climb  rate  is  10  fps
greater than abs
             %value of TCAS vertspeed
             if ClimbRate <TCAS_vertspeed -10
                 Comp_A = 1;
             else
                 Comp_A = 0;
             end

         elseif  alerttype_1 == 6
             if alerttype_5 ==0
                %Set rate required to comply
                 if alerttype_4== 1
                         Comply_Rate = 0 ;          %Climb
                 elseif alerttype_4 == 2
                         Comply_Rate = 0 ;           %Do not
descend
                 elseif alerttype_4 == 3
                         Comply_Rate  =  -500/60  ;        %To
comply to the advisory, do not descend greater than 500 fpm
                 elseif alerttype_4 == 4
                         Comply_Rate  =  -1000/60  ;       %To
comply  to  the  advisory,  do  not  descend  greater  than  1000
fpm
                 elseif alerttype_4 == 5
                         Comply_Rate = -2000/60 ;
                 else
                     error                   %To comply to the
advisory, do not descend greater than 2000 fpm
                 end
                 %Check for compliance
                 if  ClimbRate  <  Comply_Rate     %if  pilot
vertical speed (ClimbRate) is greater than compliance rate,
pilot is not in compliance
                     Comp_Rate_0 = 0;         %Comp_Rate_0 =
0 == for no buffer, the pilot is NOT IN COMPLIANCE
                 else
                     Comp_Rate_0 = 1;        %in Compliance
                 end
```

```
                if ClimbRate < Comply_Rate -0.5   %0.5 fps
buffer
                    Comp_Rate_5 = 0;         %Comp_Rate_5 =
0 == for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_5 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -1 %1 fps buffer
                    Comp_Rate_1 = 0;         %Comp_Rate_1 =
0 == for 1 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_1 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -2      %2 fps
buffer
                    Comp_Rate_2 = 0;         %Comp_Rate_2 =
0 == for 2 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_2 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -3    %3 fps
buffer
                    Comp_Rate_3 = 0;         %Comp_Rate_3 =
0 == for 3 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_3 = 1;        %in Compliance
                end

                if ClimbRate < Comply_Rate -10
                    Comp_A = 1;
                else
                    Comp_A = 0;
                end

            elseif alerttype_4 ==0
               %Set rate required to comply
                if alerttype_5 == 1
                        Comply_Rate       =       0       ;
%descend
                elseif alerttype_5 == 2
                        Comply_Rate = 0 ;           %Do not
climb
                elseif alerttype_5 == 3
```

```
                        Comply_Rate  =  500/60  ;          %To
comply to the advisory, do not climb greater than 500 fpm
                elseif alerttype_5 == 4
                        Comply_Rate  =  1000/60  ;         %To
comply to the advisory, do not climb greater than 1000 fpm
                elseif alerttype_5 == 5
                        Comply_Rate  =  2000/60  ;         %To
comply to the advisory, do not climb greater than 2000 fpm
                else
                    error
                end
                %Check for compliance
                if  ClimbRate  >  Comply_Rate     %if  pilot
vertical  speed  (ClimbRate)  is  less  than  compliance  rate,
pilot is not in compliance
                    Comp_Rate_0 = 0;          %Comp_Rate_0 =
0 == for no buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_0 = 1;        %in Compliance
                end

                if ClimbRate > Comply_Rate  +0.5  %0.5 fps
buffer
                    Comp_Rate_5 = 0;          %Comp_Rate_5 =
0 == for 0.5 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_5 = 1;        %in Compliance
                end

                if  ClimbRate  >  Comply_Rate    +1  %1  fps
buffer
                    Comp_Rate_1 = 0;          %Comp_Rate_1 =
0 == for 1 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_1 = 1;        %in Compliance
                end

                if ClimbRate > Comply_Rate   +2     %2  fps
buffer
                    Comp_Rate_2 = 0;          %Comp_Rate_2 =
0 == for 2 fps buffer, the pilot is NOT IN COMPLIANCE
                else
                    Comp_Rate_2 = 1;        %in Compliance
                end

                if  ClimbRate  >  Comply_Rate    +3    %3  fps
buffer
```

```
                Comp_Rate_3 = 0;          %Comp_Rate_3 =
0 == for 3 fps buffer, the pilot is NOT IN COMPLIANCE
            else
                Comp_Rate_3 = 1;        %in Compliance
            end

            if ClimbRate > Comply_Rate +10
                Comp_A = 1;
            else
                Comp_A = 0;
            end

        elseif alerttype_1 == 1
        Comp_A = 0;
        Comp_Rate_0 = 1;
        Comp_Rate_5 = 1;
        Comp_Rate_1 = 1;
        Comp_Rate_2 = 1;
        Comp_Rate_3 = 1;
        else
            error
        end
    end
else
        Comp_A = 0;
        Comp_Rate_0 = 1;
        Comp_Rate_5 = 1;
        Comp_Rate_1 = 1;
        Comp_Rate_2 = 1;
        Comp_Rate_3 = 1;
end





%If the RA type changes, add "hold time"
if TCAS_num(i,8) == TCAS_num(i-1,8)
    Change_RA = 0;
    RA_vert_orig = TCAS_num(i,8);
    RA_vert_new = 0;
elseif TCAS_num(i,8) ~= TCAS_num(i-1,8)
    Change_RA = 1;
    Holdtime_new = TCAS_num(i,1)+2.5;
```

```
        if Holdtime_new>Holdtime;         %If the RA resets
before the original holdtime ends then keep larger hold
time.
            Holdtime = Holdtime_new;
        else
            Holdtime = Holdtime;
        end
            if TCAS_num(i,8) ~= -1
                RA_vert_orig = TCAS_num(i-1,8);
                RA_vert_new = TCAS_num(i,8);
            end
    end

    %record values
    new_row = {time, ClimbRate, Vert_Att, Altitude,
TurnRate, Heading, Pilot_Response_Horiz, [],
TCAS_vertspeed, alert_output, [], Pilot_H , Comp_A,
Comp_Rate_0,Comp_Rate_5,Comp_Rate_1,Comp_Rate_2,Comp_Rate_3
, AP_OnOff,[],[], alert_output_numeric};

    %Compile over time by adding new rows
    outputs = [outputs; new_row];

end




%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%Eyetracker%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%

%Reads through eyetracker data and adds to the RFS data.
Since the time
%stamps do not match exactly between RFS and the
eyetracker, the eyetracker
%data is interpolated for the RFS (ie, if the eyetracker
records something
%at 121.12 and RFS records something at 121,24 then the
eyetracker will
%record their 121.12 finding into the 121.24 RFS data to
keep the data
%consistent.
```

```
new_row={};
run=true;
i=2;
time_RA=TCAS_raw{RA_start,1};  %Time RA occurs

%Search through the eyetracker data and capture data that
was recorded
%during the RA
while run
    if ET_raw{i,2}>= time_RA & ET_raw{i,2}<= time_COC

        new_row = {ET_raw{i,2}, [], [], [], [], [], [],
ET_raw{i,8}, [], [], [], [], [], [],[],[],[],[], [],[],[],
[]};     %feed in eyetracker data
        outputs=[outputs; new_row];

    elseif ET_raw{i,2} > time_COC
        run=false;
    end

    i=i+1;
end



%Combines the RFS and eyetracker data into one file and
sorts it
%ascending. Once the data is combined and sorted, the
eyetracker data is
%interpolated onto the RFS data.
[sorted, index]=sort([outputs{:,1}]);
outputs=outputs(index,:);
[datarows, datacolumns] = size(outputs);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs(:,8));

while run

    if emptyIndex(i,1)== 1    %checks for empty cells,
indicating that RFS did not record durring that time.
        p = i-1;
        outputs(i,8)   =   outputs(p,8);   %replaces   the
eyetracker data into the blank RFS cell
    else
        outputs(i,8)=outputs(i,8);
```

```
    end
    i=i+1;

    if i > datarows
        run = false;
    else
        run = true;
    end

end


run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new(j,:) = outputs(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%Autopilot%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%

%Since autopilot data is not continuous, the code has to
recognize
% that the last known mode would have been set prior to RA.
If the mode is
%NOT changed within the advisory, then the mode is the same
as before the
%advisory started. Must hand input last known mode in the
excel file.
```

```
new_row={};
run=true;
i=1;
Mode = 'TA';
Mode_num = 10;
[Events_rows, Events_cols] = size(Events_raw);
Mode_vert = 'TA';
Mode_num_vert= 30;
Mode_lat = 'TA';
Mode_num_lat= 29;
while run
    i=i+1;
  if i > Events_rows
         run=false;

   elseif  Events_raw{i,1}  >=  time_COC    %ET_raw{i,2}  >
time_RA
        run=false;


%     elseif  Events_raw{i,1}>= time_RA & Events_raw{i,1}<=
time_COC
    elseif  Events_raw{i,1}<= time_COC
          if Events_raw{i,2} == 26
               Mode = 'FLCH';
            Mode_vert = Mode;
            Mode_num_vert = 26;
          elseif Events_raw{i,2} == 27
               Mode = 'AltHold';
            Mode_vert = Mode;
            Mode_num_vert = 27;
          elseif Events_raw{i,2} == 28
               Mode = 'VS';
            Mode_vert = Mode;
            Mode_num_vert = 28;
          elseif Events_raw{i,2} == 29
               Mode = 'LNAV';
            Mode_lat = Mode;
            Mode_num_lat = 29;
          elseif Events_raw{i,2} == 30
               Mode = 'VNAV';
            Mode_vert = Mode;
            Mode_num_vert = 30;
          elseif Events_raw{i,2} == 31
               Mode = 'Spd';
            Mode_lat = Mode;
```

```
            Mode_num_lat = 31;
          elseif Events_raw{i,2} == 32
               Mode = 'HdgHld';
            Mode_lat = Mode;
            Mode_num_lat = 32;
          elseif Events_raw{i,2} == 33
               Mode = 'HdgSlt';
            Mode_lat = Mode;
            Mode_num_lat = 33;
        else
            Mode = 'Null';
          end
        new_row = {Events_raw{i,1}, [], [], [], [], [], [],
[], [], [], [], [], [], [],[],[],[],[], [],Events_raw{i,2},
Mode, []};

        outputs_new=[outputs_new; new_row];
  end


end

LastMode = Mode;
LastMode_num = Mode_num;

% The following code sorts the RFS and Mode Change data and
outputs
% the data file, similar to how the eyetracker sorts the
data.
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,20));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,20) = outputs_new(j,20);
        outputs_new(j-1,21) = outputs_new(j,21);
    end
end



run = true;
```

```
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;

while run
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end
    i=i+1;
    if i > datarows
        run = false;
    else
        run = true;
    end

end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Audio%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
% The following code sorts the RFS and AP data and outputs
the data file,
% similar to how the eyetracker sorts the data.
new_row={};
run=true;
i=1;
time_RA=TCAS_raw{RA_start,1};
[audio_rows, audio_cols] = size(audio_scaled);

while run
    if   audio_scaled{i,1}>=  time_RA  &  audio_scaled{i,1}<=
time_COC
        new_row  =  {audio_scaled{i,1},  [],  [],  [],  [],  [],
[], audio_scaled{i,2}, [], [], [], [], [], [],[],[],[],[],
[],[],[], []};
        outputs_new=[outputs_new; new_row];

    elseif audio_scaled{i,1} >= time_COC
        run=false;

    end
    i=i+1;
```

```
    if i> audio_rows
        run = false;
    end
end



% The following code sorts the RFS and audio data and
outputs the data file
[sorted, index]=sort([outputs_new{:,1}]);
outputs_new=outputs_new(index,:);
[datarows, datacolumns] = size(outputs_new);
run = true;
i=2;
emptyIndex = cellfun(@isempty,outputs_new(:,8));

for j = 2:datarows
    check = emptyIndex(j,1)-emptyIndex(j-1,1);
    if check < 0
        outputs_new(j-1,8) = outputs_new(j,8);
    end
end



run = true;
i=1;
emptyIndex = cellfun(@isempty,outputs_new(:,2));
j = 1;
[datarows, datacolumns] = size(outputs_new);
for i = 1:datarows
    if emptyIndex(i,1)== 0
       outputs_new2(j,:) = outputs_new(i,:);
       j = j+1;
    end

end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%Saving                          to
Excel%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
```

```
outputs_new2=['time','PilotClimbRate','PilotVerticalRespons
e',       'Altitude','PilotTurnRate',       'PilotHeading',
'PilotHorizontalResponse',ET_raw(1,8),     'TCAS_vertspeed',
'RA_Type',    'Audio',    'Horizontal',    'Agressiveness',
'Compliance_Rate_0','Compliance_Rate_5',
'Compliance_Rate_1','Compliance_Rate_2','Compliance_Rate_3'
,'AutoPilot','NumericMode','Mode','RATypeNumeric';
outputs_new2];
RA_outputs = outputs_new2;

%%Record the data to a text file. The excel function is
really crappy for
%%this code, for an unknown reason. Must use fprintf
instead of xlswrite.

filename_data=['./',dir_name,'/',filename, '_Outputs.txt'];

fid=fopen(filename_data,'w');
for i = 1:length(outputs_new2(:,1))
    for j = 1:length(outputs_new2(1,:))
        k = class(outputs_new2{i,j});
        if isempty(outputs_new2{i,j})
            fprintf(fid, 'empty,');
        elseif strcmp(k,'double')
            fprintf(fid, '%f,',outputs_new2{i,j});
        elseif strcmp(k,'str')
            fprintf(fid, '%f,',outputs_new2{i,j});
        elseif strcmp(k,'char')
            fprintf(fid, '%s,',outputs_new2{i,j});
        elseif strcmp(k,'cell')
            fprintf(fid, '%s,',outputs_new2{i,j});
        else
            fprintf(fid, '%s,',outputs_new2{i,j});
        end
    end
        fprintf(fid, '\n');
end
fclose(fid);




%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%Summarizing                        into
Statistics%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%% Checking for compliance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%

time_RA=TCAS_raw{RA_start,1};   %start at time of RA
RA_DURATION = time_COC-time_RA ;    %RA duration

[rows, columns] = size(RA_outputs);

%initially set everything equal to 0
RA_APOff = 0;
RA_CallATC = 0;
RA_Horiz_Man =0;
RA_Vert_Comp = 0;
RA_Agressive = 0;
RA_diff = [];
pilot_rate = [];
RA_Rate_0 = 0;
RA_Rate_1 = 0;
RA_Rate_2 = 0;
RA_Rate_5 = 0;
RA_Rate_3 = 0;

%%
% RA_outputs = {time, ClimbRate, Vert_Att, Altitude,
TurnRate, Heading, Pilot_Response_Horiz, [],
TCAS_vertspeed, alert_output, [], Pilot_H , Comp_A,
Comp_Rate_0,Comp_Rate_5,Comp_Rate_1,Comp_Rate_2,Comp_Rate_3
, AP_OnOff,[],[], alert_output_numeric};

%Exactly matched vertical rate NO BUFFER
time_comp_start_0 = {time_RA};
time_comp_stop_0 = {};

time_comp_start_5 = {time_RA};
time_comp_stop_5 = {};

time_comp_start_1 = {time_RA};
time_comp_stop_1 = {};

time_comp_start_2 = {time_RA};
```

```
time_comp_stop_2 = {};

time_comp_start_3 = {time_RA};
time_comp_stop_3 = {};


%for the entire file (1-rows), looking for if the pilot
complied.
for i = 3:rows

    pilot_rate = [pilot_rate; RA_outputs{i,2}]; %col 2 ==
pilot vert speed
    difference =  RA_outputs{i,2}-RA_outputs{i,9};  %col
9== TCAS_Vert Speed, this takes the difference from the
TCAS vert speed and the pilot's vert speed
    RA_diff = [RA_diff; difference];    %saves the vert
speed difference


%looking for where the bit flips from a 1 to a 0 reason why
we do it
% this way: for the first 5 seconds the pilot is always in
compliance
% due to RA logic (assumes 5 sec response time).Looking for
the bit flip
% tells us how long after, including those 5 secs the pilot
took to comply.
% If the pilot started in compliance and remained in
compliance then we can
% see that as well.
%if the pilot was complying (1) and then stopped complying
(0) then
% changes = -1 (<0). If the pilot was not complying (1) and
then started
% complying, then changes = 1 (>0)
    changes_0 = RA_outputs{i,14} - RA_outputs{i-1,14} ;
%comply current row with no buffer - comply previous row
with no buffer
    changes_5 = RA_outputs{i,15} - RA_outputs{i-1,15} ;
%comply current row with .5fps buffer - comply previous row
with .5fps buffer
    changes_1 = RA_outputs{i,16} - RA_outputs{i-1,16} ;
%comply current row with 1fps buffer - comply previous row
with 1fps buffer
    changes_2 = RA_outputs{i,17} - RA_outputs{i-1,17} ;
%comply current row with 2fps buffer - comply previous row
with 2fps buffer
```

```
    changes_3 = RA_outputs{i,18} - RA_outputs{i-1,18} ; %comply current row with 3fps buffer - comply previous row with 3fps buffer



    if changes_0 > 0 %looks for when the pilot was complying and then stopped complying
            time_comp_start_0 = [time_comp_start_0; RA_outputs{i,1}];

    elseif changes_0 < 0 %looks for when the pilot was not complying and then started complying
           time_comp_stop_0 = [time_comp_stop_0; RA_outputs{i,1}];
    end


    if changes_5 > 0 %looks for when the pilot was complying and then stopped complying
            time_comp_start_5 = [time_comp_start_5; RA_outputs{i,1}];

    elseif changes_5 < 0 %looks for when the pilot was not complying and then started complying
           time_comp_stop_5 = [time_comp_stop_5; RA_outputs{i,1}];
    end

    if changes_1 > 0 %looks for when the pilot was complying and then stopped complying
            time_comp_start_1 = [time_comp_start_1; RA_outputs{i,1}];

    elseif changes_1 < 0 %looks for when the pilot was not complying and then started complying
           time_comp_stop_1 = [time_comp_stop_1; RA_outputs{i,1}];
    end

     if changes_2 > 0 %looks for when the pilot was complying and then stopped complying
            time_comp_start_2 = [time_comp_start_2; RA_outputs{i,1}];
```

```
    elseif changes_2 < 0 %looks for when the pilot was not
complying and then started complying
           time_comp_stop_2      =      [time_comp_stop_2;
RA_outputs{i,1}];
     end

    if changes_3 > 0 %looks for when the pilot was
complying and then stopped complying
            time_comp_start_3     =     [time_comp_start_3;
RA_outputs{i,1}];

     elseif changes_3 < 0 %looks for when the pilot was not
complying and then started complying
           time_comp_stop_3      =      [time_comp_stop_3;
RA_outputs{i,1}];
    end



end

RA_diff_average = mean(RA_diff);    %Average vertical speed
difference
RA_diff_max = max(abs(RA_diff),[],1);   %max vertical speed
difference
pilot_rate_max  =  max(abs(pilot_rate),[],1);          %max
vertical speed

abs_RA_diff_average = abs(RA_diff_average);


%%%%%%%%%%%%%%Percentage   Compliance   with   NO   Buffer
%%%%%%%%%%%%%%%%%%%%%%
 [starttimes_0, dummy] = size(time_comp_start_0);
 [stoptimes_0, dummy] = size(time_comp_stop_0);

Compliance_Block_0 = [];

for i=1:(starttimes_0-1)
    Compliance_Block_0        =        [Compliance_Block_0;
time_comp_stop_0{i,1}-time_comp_start_0{i,1}];
end

if starttimes_0 == stoptimes_0
    Compliance_Block_0        =        [Compliance_Block_0;
time_comp_stop_0{end,1}-time_comp_start_0{end,1}];
elseif starttimes_0 > stoptimes_0
```

```
   Compliance_Block_0   =   [Compliance_Block_0;   time_COC-
time_comp_start_0{end,1}];
else
    error
end

Compliance_Total_Time_0 = sum(Compliance_Block_0);
percentage_compliance_0                                      =
Compliance_Total_Time_0/RA_DURATION     *100;  %time  pilot
complied/total time of RA == percentage compliance

if starttimes_0 > 1
    RA_MatchRate_0 = time_comp_start_0{2,1}-time_RA;
elseif starttimes_0 == 1
    if stoptimes_0 == 0
        RA_MatchRate_0 = 0 ;
    elseif stoptimes_0 == 1
        RA_MatchRate_0 = -2;
    else
        error
    end
end

if percentage_compliance_0 == 100
    RA_Rate_0 =1;
else
    RA_Rate_0 =0;
end

%%%%%%%%%%%%%Percentage  Compliance  with  0.5  FPS  Buffer
%%%%%%%%%%%%%%%%%%%%%
 [starttimes_5, dummy] = size(time_comp_start_5);
 [stoptimes_5, dummy] = size(time_comp_stop_5);

Compliance_Block_5 = [];

for i=1:(starttimes_5-1)
    Compliance_Block_5        =        [Compliance_Block_5;
time_comp_stop_5{i,1}-time_comp_start_5{i,1}];
end

if starttimes_5 == stoptimes_5
    Compliance_Block_5        =        [Compliance_Block_5;
time_comp_stop_5{end,1}-time_comp_start_5{end,1}];
elseif starttimes_5 > stoptimes_5
   Compliance_Block_5   =   [Compliance_Block_5;   time_COC-
time_comp_start_5{end,1}];
```

```
else
    error
end

Compliance_Total_Time_5 = sum(Compliance_Block_5);
percentage_compliance_5                                          =
Compliance_Total_Time_5/RA_DURATION     *100;   %time   pilot
complied/total time of RA == percentage compliance

if starttimes_5 > 1
    RA_MatchRate_5 = time_comp_start_5{2,1}-time_RA;
elseif starttimes_5 == 1
    if stoptimes_5 == 0
        RA_MatchRate_5 = 0 ;
    elseif stoptimes_5 == 1
        RA_MatchRate_5 = -2;
    else
        error
    end
end

if percentage_compliance_5 == 100
    RA_Rate_5 =1;
else
    RA_Rate_5 =0;
end
%%%%%%%%%%%%%Percentage   Compliance   with   1   FPS   Buffer
%%%%%%%%%%%%%%%%%%%%%
 [starttimes_1, dummy] = size(time_comp_start_1);
 [stoptimes_1, dummy] = size(time_comp_stop_1);

Compliance_Block_1 = [];

for i=1:(starttimes_1-1)
    Compliance_Block_1        =        [Compliance_Block_1;
time_comp_stop_1{i,1}-time_comp_start_1{i,1}];
end

if starttimes_1 == stoptimes_1
    Compliance_Block_1        =        [Compliance_Block_1;
time_comp_stop_1{end,1}-time_comp_start_1{end,1}];
elseif starttimes_1 > stoptimes_1
   Compliance_Block_1   =   [Compliance_Block_1;   time_COC-
time_comp_start_1{end,1}];
else
    error
end
```

```
Compliance_Total_Time_1 = sum(Compliance_Block_1);
percentage_compliance_1 =
Compliance_Total_Time_1/RA_DURATION *100; %time pilot
complied/total time of RA == percentage compliance


if starttimes_1 > 1
    RA_MatchRate_1 = time_comp_start_1{2,1}-time_RA;
elseif starttimes_1 == 1
    if stoptimes_1 == 0
        RA_MatchRate_1 = 0 ;
    elseif stoptimes_1 == 1
        RA_MatchRate_1 = -2;
    else
        error
    end
end

if percentage_compliance_1 == 100
    RA_Rate_1 =1;
else
    RA_Rate_1 =0;
end
%%%%%%%%%%%%%%Percentage Compliance with 2 FPS Buffer
%%%%%%%%%%%%%%%%%%%%%%
 [starttimes_2, dummy] = size(time_comp_start_2);
 [stoptimes_2, dummy] = size(time_comp_stop_2);

Compliance_Block_2 = [];

for i=1:(starttimes_2-1)
    Compliance_Block_2 = [Compliance_Block_2;
time_comp_stop_2{i,1}-time_comp_start_2{i,1}];
end

if starttimes_2 == stoptimes_2
    Compliance_Block_2 = [Compliance_Block_2;
time_comp_stop_2{end,1}-time_comp_start_2{end,1}];
elseif starttimes_2 > stoptimes_2
   Compliance_Block_2 = [Compliance_Block_2; time_COC-
time_comp_start_2{end,1}];
else
    error
end

Compliance_Total_Time_2 = sum(Compliance_Block_2);
```

```
percentage_compliance_2                                    =
Compliance_Total_Time_2/RA_DURATION     *100;  %time  pilot
complied/total time of RA == percentage compliance

if starttimes_2 > 1
    RA_MatchRate_2 = time_comp_start_2{2,1}-time_RA;
elseif starttimes_2 == 1
    if stoptimes_2 == 0
        RA_MatchRate_2 = 0 ;
    elseif stoptimes_2 == 1
        RA_MatchRate_2 = -2;
    else
        error
    end
end

if percentage_compliance_2 == 100
    RA_Rate_2 =1;
else
    RA_Rate_2 =0;
end
%%%%%%%%%%%%%Percentage  Compliance  with  3  FPS  Buffer
%%%%%%%%%%%%%%%%%%%%%
 [starttimes_3, dummy] = size(time_comp_start_3);
 [stoptimes_3, dummy] = size(time_comp_stop_3);

Compliance_Block_3 = [];

for i=1:(starttimes_3-1)
    Compliance_Block_3        =        [Compliance_Block_3;
time_comp_stop_3{i,1}-time_comp_start_3{i,1}];
end

if starttimes_3 == stoptimes_3
    Compliance_Block_3        =        [Compliance_Block_3;
time_comp_stop_3{end,1}-time_comp_start_3{end,1}];
elseif starttimes_3 > stoptimes_3
   Compliance_Block_3   =   [Compliance_Block_3;   time_COC-
time_comp_start_3{end,1}];
else
    error
end

Compliance_Total_Time_3 = sum(Compliance_Block_3);
percentage_compliance_3                                    =
Compliance_Total_Time_3/RA_DURATION     *100;  %time  pilot
complied/total time of RA == percentage compliance
```

```
if starttimes_3 > 1
    RA_MatchRate_3 = time_comp_start_3{2,1}-time_RA;
elseif starttimes_3 == 1
    if stoptimes_3 == 0
        RA_MatchRate_3 = 0 ;
    elseif stoptimes_3 == 1
        RA_MatchRate_3 = -2;
    else
        error
    end
end

if percentage_compliance_3 == 100
    RA_Rate_3 =1;
else
    RA_Rate_3 =0;
end

%% Binary values and Others are set below this line

[rows, columns] = size(RA_outputs);
[AP_rows, AP_cols] = size(AP_raw);


alt1 = RA_outputs{2,4};
alt2 = RA_outputs{rows,4};
AltitudeChange = alt2 - alt1;

hdg1 = RA_outputs{2,6};
hdg2 = RA_outputs{rows,6};
HeadingChange = hdg2-hdg1;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
%%%%%%%%%%%%%          Binary          Horizontal          and
Aggressive%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
if HeadingChange >=15
    RA_Horiz_Man = 1;
end

skip_H = 0;
skip_V = 0;
skip_A = 0;
```

```
skip_0 = 0;
skip_5 = 0;
skip_1 = 0;
skip_2 = 0;
skip_3 = 0;



for i = 2:rows;

    RA_Horiz = RA_outputs{i, 12};
    if RA_Horiz == 1 && skip_H == 0;
        RA_Horiz_Man = 1;
        skip_H = 1;
    elseif RA_Horiz_Man ~= 1 && skip_H == 0;
        RA_Horiz_Man = 0;
    end

      RA_Ag = RA_outputs{i, 13};
    if RA_Ag == 1 && skip_A ==0;
        RA_Agressive = 1;
        skip_A= 1;
    elseif RA_Agressive ~= 1 && skip_A ==0;
        RA_Agressive = 0;

    end




end



%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%AutoPilot%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%did the pilot turn their AP off? And, at what time?
run = true;
i = 2;
while run
    RA_AP =  RA_outputs{i,19};
    if RA_AP == 0;
        RA_APOff = 1;
        RA_APOff_time = RA_outputs{i,1}-RA_outputs{2,1};
```

```
        run = false;
    else
        RA_APOff = 0;

    end
    i = i+1;
    if i>rows
        run = false;
    end
end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%Call ATC%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Call ATC? What time?

run = true;
i = 2;
while run

    check = strcmp('PTTS', RA_outputs{i,11});
    if check == 1;
        RA_CallATC = 1;
        RA_CallATC_time = RA_outputs{i,1}-RA_outputs{2,1};
        run = false;
    else
        if RA_CallATC ~=1
        RA_CallATC = 0;
        end
    end

    i = i+1;
    if i>rows
        run = false;
    end
end

if RA_CallATC == 0;
    RA_CallATC_time = -2;
end

if RA_APOff == 0;
    RA_APOff_time = -2;
end
```

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%Eyetracker%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Did the pilot look at a specific display? For how long
over duration of event?
     check_ND = [RA_outputs{2,1}, strcmp('ND',
RA_outputs{2,8})];
     check_TSD = [RA_outputs{2,1}, strcmp('TSD',
RA_outputs{2,8})];
     check_PFD = [RA_outputs{2,1}, strcmp('PFD',
RA_outputs{2,8})];
     check_CDU = [RA_outputs{2,1}, strcmp('CDU',
RA_outputs{2,8})];
     check_MCP = [RA_outputs{2,1}, strcmp('MCP',
RA_outputs{2,8})];


     check_ND_starttime=[];
     check_TSD_starttime=[];
     check_PFD_starttime=[];
     check_CDU_starttime=[];
     check_MCP_starttime=[];
     check_ND_stoptime = [];
     check_TSD_stoptime = [];
     check_PFD_stoptime = [];
     check_MCP_stoptime = [];
     check_CDU_stoptime = [];

     %Record if the pilot looked at the display
     if check_ND(1,2) == 1
         check_ND_starttime = check_ND(1,1);
         size_ND = 1;
     end

     if check_TSD(1,2) == 1
         check_TSD_starttime = check_TSD(1,1);
     end

     if check_PFD(1,2) == 1
         check_PFD_starttime = check_PFD(1,1);
     end

     if check_CDU(1,2) == 1
```

```
            check_CDU_starttime = check_CDU(1,1);
        end

        if check_MCP(1,2) == 1
            check_MCP_starttime = check_MCP(1,1);
        end

for i = 3:rows


     check_ND  =  [check_ND;  RA_outputs{i,1},  strcmp('ND',
RA_outputs{i,8})];
     check_TSD = [check_TSD; RA_outputs{i,1}, strcmp('TSD',
RA_outputs{i,8})];
     check_PFD = [check_PFD; RA_outputs{i,1}, strcmp('PFD',
RA_outputs{i,8})];
     check_CDU = [check_CDU; RA_outputs{i,1}, strcmp('CDU',
RA_outputs{i,8})];
     check_MCP = [check_MCP; RA_outputs{i,1}, strcmp('MCP',
RA_outputs{i,8})];

end

for i = 2:rows-2

    if check_ND(i,2)-check_ND(i-1,2) >0
         check_ND_starttime      =      [check_ND_starttime;
check_ND(i,1)];
    elseif check_ND(i,2)-check_ND(i-1,2) <0
         check_ND_stoptime       =       [check_ND_stoptime;
check_ND(i,1)];
    end

    if check_TSD(i,2)-check_TSD(i-1,2) >0
         check_TSD_starttime     =    [check_TSD_starttime;
check_TSD(i,1)];
    elseif check_TSD(i,2)-check_TSD(i-1,2) <0
         check_TSD_stoptime      =     [check_TSD_stoptime;
check_TSD(i,1)];
    end

    if check_PFD(i,2)-check_PFD(i-1,2) >0
         check_PFD_starttime     =    [check_PFD_starttime;
check_PFD(i,1)];
    elseif check_PFD(i,2)-check_PFD(i-1,2) <0
         check_PFD_stoptime      =     [check_PFD_stoptime;
check_PFD(i,1)];
```

```
    end

    if check_CDU(i,2)-check_CDU(i-1,2) >0
         check_CDU_starttime     =    [check_CDU_starttime;
check_CDU(i,1)];
    elseif check_CDU(i,2)-check_CDU(i-1,2) <0
         check_CDU_stoptime      =     [check_CDU_stoptime;
check_CDU(i,1)];
    end

    if check_MCP(i,2)-check_MCP(i-1,2) >0
         check_MCP_starttime     =    [check_MCP_starttime;
check_MCP(i,1)];
    elseif check_MCP(i,2)-check_MCP(i-1,2) <0
         check_MCP_stoptime      =     [check_MCP_stoptime;
check_MCP(i,1)];
    end
end

[check_ND_starttime_rows,   check_ND_starttime_columns]   =
size(check_ND_starttime);
[check_ND_stoptime_rows,    check_ND_stoptime_columns]    =
size(check_ND_stoptime);
[check_TSD_starttime_rows,  check_TSD_starttime_columns]  =
size(check_TSD_starttime);
[check_TSD_stoptime_rows,   check_TSD_stoptime_columns]   =
size(check_TSD_stoptime);
[check_PFD_starttime_rows,  check_PFD_starttime_columns]  =
size(check_PFD_starttime);
[check_PFD_stoptime_rows,   check_PFD_stoptime_columns]   =
size(check_PFD_stoptime);
[check_CDU_starttime_rows,  check_CDU_starttime_columns]  =
size(check_CDU_starttime);
[check_CDU_stoptime_rows,   check_CDU_stoptime_columns]   =
size(check_CDU_stoptime);
[check_MCP_starttime_rows,  check_MCP_starttime_columns]  =
size(check_MCP_starttime);
[check_MCP_stoptime_rows,   check_MCP_stoptime_columns]   =
size(check_MCP_stoptime);



if check_ND_starttime_rows>check_ND_stoptime_rows
    check_ND_stoptime         =         [check_ND_stoptime;
check_ND(end,1)];
end
```

```
if check_TSD_starttime_rows>check_TSD_stoptime_rows
    check_TSD_stoptime        =        [check_TSD_stoptime;
check_TSD(end,1)];
end

if check_PFD_starttime_rows>check_PFD_stoptime_rows
    check_PFD_stoptime        =        [check_PFD_stoptime;
check_PFD(end,1)];
end

if check_CDU_starttime_rows>check_CDU_stoptime_rows
    check_CDU_stoptime        =        [check_CDU_stoptime;
check_CDU(end,1)];
end


if check_MCP_starttime_rows>check_MCP_stoptime_rows
    check_MCP_stoptime        =        [check_MCP_stoptime;
check_MCP(end,1)];
end


duration_ND_temp=[];
duration_TSD_temp=[];
duration_PFD_temp=[];
duration_CDU_temp=[];
duration_MCP_temp=[];



for i = 1:check_ND_starttime_rows
    duration_ND_temp          =          [duration_ND_temp;
check_ND_stoptime(i,1)-check_ND_starttime(i,1)];
end

for i = 1:check_TSD_starttime_rows
    duration_TSD_temp         =         [duration_TSD_temp;
check_TSD_stoptime(i,1)-check_TSD_starttime(i,1)];
end

for i = 1:check_PFD_starttime_rows
    duration_PFD_temp         =         [duration_PFD_temp;
check_PFD_stoptime(i,1)-check_PFD_starttime(i,1)];
end

for i = 1:check_CDU_starttime_rows
```

```
    duration_CDU_temp          =          [duration_CDU_temp;
check_CDU_stoptime(i,1)-check_CDU_starttime(i,1)];
end


for i = 1:check_MCP_starttime_rows
    duration_MCP_temp          =          [duration_MCP_temp;
check_MCP_stoptime(i,1)-check_MCP_starttime(i,1)];
end

TOTAL_ND_VIEWING = sum(duration_ND_temp);
TOTAL_TSD_VIEWING = sum(duration_TSD_temp);
TOTAL_PFD_VIEWING = sum(duration_PFD_temp);
TOTAL_CDU_VIEWING = sum(duration_CDU_temp);
TOTAL_MCP_VIEWING = sum(duration_MCP_temp);

%Binary display values
if TOTAL_ND_VIEWING == 0
    ND_View_YesNo = 0;
else

    ND_View_YesNo = 1;
end


if TOTAL_TSD_VIEWING == 0
    TSD_View_YesNo = 0;
else

    TSD_View_YesNo = 1;
end

if TOTAL_PFD_VIEWING == 0
    PFD_View_YesNo = 0;
else

    PFD_View_YesNo = 1;
end

if TOTAL_CDU_VIEWING == 0
    CDU_View_YesNo = 0;
else

    CDU_View_YesNo = 1;
end

if TOTAL_MCP_VIEWING == 0
```

```
    MCP_View_YesNo = 0;
else

    MCP_View_YesNo = 1;
end



duration_RA = RA_outputs{rows,1}- RA_outputs{2,1};
time_RA=TCAS_raw{RA_start,1};
RA_DURATION = time_COC-time_RA ;


absAltChange = abs(AltitudeChange);

%%
RAType = cell2mat(RA_outputs(2,22));

%%

%OUTPUTS!!%
RA_Statistics     =     [RAType,     RA_APOff,RA_APOff_time,
RA_CallATC,RA_CallATC_time,      RA_Horiz_Man,HeadingChange,
RA_Rate_0,RA_Rate_5,          RA_Rate_1,RA_Rate_2,RA_Rate_3,
RA_MatchRate_0,     RA_MatchRate_5,          RA_MatchRate_1,
RA_MatchRate_2,                              RA_MatchRate_3,
RA_Agressive,abs_RA_diff_average,               RA_diff_max,
pilot_rate_max,
absAltChange,percentage_compliance_0,percentage_compliance_
5,percentage_compliance_1,percentage_compliance_2,percentag
e_compliance_3,         TOTAL_ND_VIEWING,TOTAL_TSD_VIEWING,
TOTAL_PFD_VIEWING,                        TOTAL_CDU_VIEWING,
TOTAL_MCP_VIEWING,ND_View_YesNo,             TSD_View_YesNo,
PFD_View_YesNo,  CDU_View_YesNo,  MCP_View_YesNo,  RA_start,
duration_RA, Mode_num_vert, Mode_num_lat];


 end
```

REFERENCES

Anderson, John. (1983). *The Architecture of Cognition*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc.

Aviation Safety Council. (2007). Far Eastern Air Transport Flight EF306, Boeing 757-200/Thai International Airways Flight TG659, Boeing 777-300 Near Colision at an Altitude of 34,000 Ft. and 99 NM South of Jeju Island, Korea on November 16, 2006. Taipei, Taiwan.

Bandura, Albert. (1986). *Social Foundations of Thought and Action: A Social Cognitive Theory*. Rockville, MD: National Institutes of Mental Health.

Coso, A., Fleming, E., & Pritchett, A. (2011). *Characterizing Pilots' Interactions with the Aircraft Collision Avoidance System*. Paper presented at the 16th International Symposium on Aviation Psychology, Dayton, OH.

Driskell, J., Salas, E., Johnston, J., & Wollert, T. (2008). Stress Exposure Training: An Event-Based Approach *Performance Under Stress* (pp. 271-286). London: Ashgate.

Dwyer, D., Oser, R., Salas, E., & Fowlkes, J. (1999). Performance Measurement in Distributed Environments: Initial Results and Implication for Training. *Military Psychology*, 11(2), 189-215.

Eurocontrol. (2011). Focus on Pilot Training (Vol. 12). ACAS II Bulletin.

Federal Aviation Administration (2011a). Advisory Circular No. 120-55C: Air Carrier Operational Approval and Use of TCAS II. Washington, D.C.

Federal Aviation Administration (2011b). Introduction to TCAS II Version 7.1. Washington, D.C.

Federal Aviation Administration (2003). Federal Aviation Regulations Part 91: Responsibility and Authority of the Pilot in Command, 91.3. Washington, D.C.

Flight Standards Service. (2011). TCAS II Guidance and Training for Title 14 of the Code of Federal Regulations (14 CFR) Part 135 & Part 125 Certificate Holders, Part 125 Letter of Deviation Authority (LODA) holders (125M), Part 91 Subpart K, (91K) Program Managers, Part 91 Operators and Part 142 Training Centers. Safety Alert for Operators.

Fowlkes, J., Dwyer, D., Oser, R., & Salas, E. (1998). Event-Based Approach to Training (EBAT). *The International Jounal of Aviation Psychology*, 8(3), 209-221.

Ivergard, Toni, & Hunt, Brian. (2009). *Handbook of Control Room Design and Ergonomics: A Perspective for the Future* (2nd ed.). Boca Raton, FL: CRC Press.

Lacagnina, M. (2008). Easy Does It. *Flight Safety Foundation AeroSafety World*.

Ladkin, P. (2004). *Causal analysis of the ACAS/TCAS sociotechnical system*. Paper presented at the Proceedings of the 9th Australian workshop on Safety critical systems and software.

Mauro, R., & Barshi, I. (2003). *Traning Smart: Using principles of cognitive science in aeronautical education and training*. Paper presented at the 41st Aerospace Sciences Meeting and Exhibit, Reno, NV.

Mellone, V. J. (1993). TCAS II, Genie out of the Bottle. *ASRS Directline* (4)

NASA. (2009). Aviation Safety Reporting System. Retrieved August 1, 2010: http://asrs.arc.nasa.gov/

Olson, W. and J. Olszta (2010). *TCAS Operational Performance Assessment in the U.S. National Airspace*. IEEE/AIAA Digital Avionics Systems Conference.

Olszta, J., & Olson, W. (2011). *Characterization and Analysis of Traffic Alert and Collision Avoidance Resolution Advisories Resulting for 500' and 1,000' Vertical Separation.* Paper presented at the Ninth USA/Europe Air Traffic Management Research and Developmen Seminar (ATM 2011), Berlin, Germany.

Olszta, J., Olson, W., McNamara, D., & Javits, L. (2011). Pilot Response to Traffic Alert and Collision Avoidance Resolution Advisories in the U.S. National Airspace System: Lincoln Laboratory, Massachusetts Institute of Technology.

Ornstein, A, & Levine, D. (1997). *Foundations in Education* (6th ed.). Boston, MA: Houghton Mifflin Company.

Petrosini, L., Grazino, A., Mandolesi, L., Neri, P., Molinari, M., & Leggio, M. (2003). Watch how we do it! New advances in learning by observation. *Brain Research Reviews*, 42(3), 252-264.

Pritchett, A., Fleming, E., Clevelance, W., Zoetrum, J., Popescu, V., & Thakkar, D. (2012a). *Pilot Interaction with TCAS and Air Traffic Control*. Paper presented at the 2nd International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS), London.

Pritchett, A., Fleming, E., Cleveland, W., Popescu, V., Thakkar, D., & Zoetrum, J. (2012b). *Pilot's Information Use During TCAS Events, and Relationship to Compliance to TCAS Resolution Advisories*. Paper presented at the 56th Annual Meeting of the Human Factors and Ergonomics, Boston, MA.

Pritchett, A., Fleming, E., Zoetrum, J., Cleveland, W., Popescu, V., & Thakkar, D. (2012c). *Examining Pilot Compliance to Collision Avoidance Advisories*. Paper presented at the 4th International Conference on Applied Human Factors and Ergonomics San Francisco.

Rasmussen, Jens. (1983). Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man and Cybernetics*, 13(3), 257-266.

Rasmussen, Jens, & Vicente, Kim. (1989). Coping with human errors through system design: implications for ecological interface design. *International Journal of Man-Machine Studies*, 31, 517-534.

Rosen, M., Salas, E., Pavlas, D., Jensen, R., Fu, D., & Lampton, D. (2010).

Demonstration Based Training: A Review of Instructional Features. *Human Factors*, 52(5), 596-609.

Schlechter, T., & Anthony, J. (1996). An Examination of the Value of Demonstration Tapes for Virtual Training Programs: US Army Research Institute for the Behavioral Social